TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Juha Ala-Luhtala

Approximate Bayesian inference methods for stochastic state space models

Julkaisu 1528 • Publication 1528

Tampere 2018

Tampereen teknillinen yliopisto. Julkaisu 1528
Tampere University of Technology. Publication 1528

Juha Ala-Luhtala

Approximate Bayesian inference methods for stochastic state space models

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Sähkötalo Building, Auditorium SA203, at Tampere University of Technology, on the 23rd of February 2018, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2018

Doctoral candidate: Juha Ala-Luhtala
Laboratory of Mathematics
Faculty of Natural Sciences
Tampere University of Technology
Finland

Supervisor: Robert Piché, Professor
Laboratory of Automation and Hydraulic Engineering
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Instructor: Simo Ali-Löytty, University Lecturer
Laboratory of Mathematics
Faculty of Natural Sciences
Tampere University of Technology
Finland

Pre-examiners: Jose A. Lopez-Salcedo, Associate Professor
Department of Telecommunications &
Systems Engineering
Universitat Autònoma de Barcelona (UAB)
Spain

Matti Vihola, Associate Professor
Department of Mathematics and Statistics
University of Jyväskylä
Finland

Opponent: Thomas Schön, Professor
Department of Information Technology
Division of Systems and Control
Uppsala University
Sweden

ISBN 978-952-15-4091-2 (printed)
ISBN 978-952-15-4106-3 (PDF)
ISSN 1459-2045

Abstract

This thesis collects together research results obtained during my doctoral studies related to approximate Bayesian inference in stochastic state-space models. The published research spans a variety of topics including 1) application of Gaussian filtering in satellite orbit prediction, 2) outlier robust linear regression using variational Bayes (VB) approximation, 3) filtering and smoothing in continuous-discrete Gaussian models using VB approximation and 4) parameter estimation using twisted particle filters. The main goal of the introductory part of the thesis is to connect the results to the general framework of estimation of state and model parameters and present them in a unified manner.

Bayesian inference for non-linear state space models generally requires use of approximations, since the exact posterior distribution is readily available only for a few special cases. The approximation methods can be roughly classified into to groups: deterministic methods, where the intractable posterior distribution is approximated from a family of more tractable distributions (e.g. Gaussian and VB approximations), and stochastic sampling based methods (e.g. particle filters). Gaussian approximation refers to directly approximating the posterior with a Gaussian distribution, and can be readily applied for models with Gaussian process and measurement noise. Well known examples are the extended Kalman filter and sigma-point based unscented Kalman filter. The VB method is based on minimizing the Kullback-Leibler divergence of the true posterior with respect to the approximate distribution, chosen from a family of more tractable simpler distributions.

The first main contribution of the thesis is the development of a VB approximation for linear regression problems with outlier robust measurement distributions. A broad family of outlier robust distributions can be presented as an infinite mixture of Gaussians, called Gaussian scale mixture models, and include e.g. the t-distribution, the Laplace distribution and the contaminated normal distribution. The VB

approximation for the regression problem can be readily extended to the estimation of state space models and is presented in the introductory part.

VB approximations can be also used for approximate inference in continuous-discrete Gaussian models, where the dynamics are modeled with stochastic differential equations and measurements are obtained at discrete time instants. The second main contribution is the presentation of a VB approximation for these models and the explanation of how the resulting algorithm connects to the Gaussian filtering and smoothing framework.

The third contribution of the thesis is the development of parameter estimation using particle Markov Chain Monte Carlo (PMCMC) method and twisted particle filters. Twisted particle filters are obtained from standard particle filters by applying a special weighting to the sampling law of the filter. The weighting is chosen to minimize the variance of the marginal likelihood estimate, and the resulting particle filter is more efficient than conventional PMCMC algorithms. The exact optimal weighting is generally not available, but can be approximated using the Gaussian filtering and smoothing framework.

Preface

The majority of the research presented in the thesis was carried out in the Positioning Algorithms group in the Tampere University of Technology between 2012 and 2015. The finalisation of the thesis was carried out between 2016 and 2017 while I was working in the IndoorAtlas corporation. The research was funded by Tampere Doctoral Programme in Information Science and Engineering (TISE) and Tampere University of Technology Doctoral Programme in Engineering and Natural Sciences. I also gratefully acknowledge additional financial support from KAUTE and Emil Aaltonen foundations.

I thank my supervisor and instructor Prof. Robert Piché for his guidance and teaching throughout my PhD journey, providing me interesting research topics and connecting me with people in the industry and academia. I thank my instructor Dr. Simo Ali-Löytty for his excellent mathematical advice, and also for tirelessly guiding me through the university bureaucracy. I'm deeply grateful to Dr. Nick Whiteley for providing me an inspiring and challenging research topics and hosting my visits to the University of Bristol during Autumn 2014 and 2015. I also thank Prof. Simo Särkkä and Dr. Kari Heine for their mentoring and collaboration. I thank all my colleagues in the Positioning Algorithms group, especially Mari Seppänen, Dr. Matti Raitoharju, Dr. Philipp Muller, Dr. Henri Pesonen and Simo Martikainen. Special thanks to Prof. Jose A. Lopez-Salzedo and Prof. Matti Vihola for pre-examination of my thesis and Prof. Thomas Schön for acting as my opponent.

Finally, I thank my family for all the support throughout my life and for always encouraging me to pursue my dreams.

Contents

Acronyms

CKF	cubature Kalman filter
CN	contaminated normal
EKF	extended Kalman filter
GHKF	Gauss-Hermite Kalman filter
GNNS	Global Navigation Satellite System
KL	Kullback-Leibler
MAP	maximum a posteriori
MCMC	Markov Chain Monte Carlo
PF	particle filter
PMCMC	particle Markov Chain Monte Carlo
RMSE	root-mean-square error
RTSS	Rauch-Tung-Striebel smoother
SDE	stochastic differential equation
UKF	unscented Kalman filter
VB	variational Bayes

Nomenclature

$x, y, \theta, \ldots$	random or non-random scalars
$\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \ldots$	random or non-random vectors
$\mathrm{F}, \mathrm{H}, \mathrm{Q}, \ldots$	matrices
$\mathrm{I}_{n \times n}$	Identity matrix of size n
$x_{a:b}$	sequence $x_a, x_{a+1}, \ldots, x_b$, with the convention that $x_{a:b}$ is an empty sequence when $b < a$
$\mathrm{N}(\mathbf{m}, \mathrm{P})$	multivariate Gaussian distribution with mean $\mathbf{m}$ and covariance matrix P
$\mathrm{N}(\mathbf{x} \,\vert\, \mathbf{m}, \mathrm{P})$	multivariate Gaussian probability density function with mean $\mathbf{m}$ and covariance matrix P
$\mathbf{x} \sim p(\mathbf{x})$	random variable $\mathbf{x}$ is distributed according to distribution or probability density $p(\mathbf{x})$
$g(\mathbf{x}) \propto p(\mathbf{x})$	the function $g(\mathbf{x})$ is proportional to function $p(\mathbf{x})$
$\dot{\mathbf{x}}(t)$	first derivative of $\mathbf{x}(t)$ w.r.t. t
$\ddot{\mathbf{x}}(t)$	second derivative of $\mathbf{x}(t)$ w.r.t. t
$\mathbb{E}[f(\mathbf{x})]$	expected value of random variable $f(\mathbf{x})$
$\mathrm{KL}[p(\mathbf{x}) \,\vert\vert\, q(\mathbf{x})]$	KL-divergence of $p(\mathbf{x})$ w.r.t. $q(\mathbf{x})$
$\nabla_{\mathbf{x}} f(\mathbf{x})$	gradient of $f(\mathbf{x})$ with respect to $\mathbf{x}$
N_{eff}	Effective sample size in particle filter algorithm
N_{thr}	Effective sample size threshold for resampling in particle filter algorithm
$n_{\max}$	Maximum iterations in outlier-robust VB filter

List of Publications

I Juha Ala-Luhtala, Mari Seppänen, Simo Ali-Löytty, Robert Piché and Henri Nurminen, Estimation of initial state and model parameters for autonomous GNSS orbit prediction, *In Proceedings of International Global Navigation Satellite Systems Society Symposium 2013 (IGNSS2013)*, 2013

II Juha Ala-Luhtala, Robert Piché, Gaussian scale mixture models for robust linear multivariate regression with missing data, *Communications in Statistics - Simulation and Computation*, 45:(3):791-813, 2016

III Juha Ala-Luhtala, Simo Särkkä, Robert Piché. Gaussian filtering and variational approximations for Bayesian smoothing in continuous-discrete stochastic dynamic systems, *Signal Processing*, 111:(0):124-136, 2015

IV Juha Ala-Luhtala, Nick Whiteley, Kari Heine, Robert Piché. An introduction to twisted particle filters and parameter estimation in non-linear state-space models, *IEEE transactions on Signal Processing*, 64:(18):4875-4890, 2016

Author's contribution

PI : I prepared the manuscript and wrote main parts of the Matlab code used to perform numerical experiments and analyse the results. Co-authors contributed in planning of the manuscript and advising during the writing process.

PII : I prepared the manuscript based on the topic suggested by of Robert Piché. I performed all the numerical experiments and analysed the results presented in the paper.

PIII : I prepared the manuscript based on the topic suggested by of Simo Särkkä. I performed the numerical experiments and analysed the results.

PIV : The topic for this paper was suggested by Nick Whiteley. Chapters I-III were jointly prepared by me and Nick Whiteley, and chapters IV-V by me. I performed all the numerical experiments and data analysis presented in the paper.

1 Introduction

1.1 Background

Probabilistic state-space models are commonly used to describe dynamical systems ranging from engineering applications to medical and biological processes [13, 33, 41, 59]. In a state-space model, the modelled system is described by a vector variable that defines the state of the system at each time instant. The transition model describes how the system's state evolves in time. Usually, the state cannot be observed directly, but only through some measurement process that is corrupted by noise. In the probabilistic setting, the transition model and measurement process are described by probability distributions.

Bayesian inference (see e.g. [14, 24, 40]) is a convenient way for doing inference in probabilistic state-space models. The components of Bayesian inference consists of a likelihood model, which links the noisy measurements to the variable of interest, and a prior distribution, which models our beliefs about the unknown variable before any measurements are observed. The prior and likelihood are combined using the Bayes theorem giving the posterior distribution, which contains all the information about the unknown variable.

In general state-space models, computing the posterior distribution is intractable and some approximations must be used. An important special case is the linear Gaussian state-space model, which allows an exact solution for the posterior distribution. The Kalman filter [31] is the well known algorithm for computing the solution recursively in the linear Gaussian case. Extensions of the Kalman filter for the nonlinear case include the extended Kalman filter, sigma-point filters (e.g. cubature Kalman filter, unscented Kalman filter, Gauss-Hermite Kalman filter) and statistically linearized filters [9, 27, 29, 49, 52]. These kind of approaches are generally called Gaussian filtering methods, since the posterior distributions are approximated as Gaussian

distributions.

A different approach for approximating the posterior distribution is using a method called Variational Bayes (VB) [15, 17]. The VB-method is based on trying to find an approximate distribution that minimizes the Kullback-Leibler divergence [38] between the approximation and the true posterior. The approximation is chosen from a certain family of distributions that yield tractable equations for the posterior. Some possible alternatives for the approximate family are factorized distributions, where the functional form of the distribution is not fixed beforehand, or fixed-form distributions e.g. Gaussian distributions, whose parameters are optimized in the minimization.

The VB-method is a useful tool for distributions that can be represented as a scale mixture of Gaussian distributions [2, 39]. These include e.g. the t-distribution [36], the Laplace distribution [37] and the contaminated normal distribution [54], all three of which are often used in outlier-robust measurement models. For state-space models, using a heavy tailed measurement distribution yields outlier-robust Kalman filtering and smoothing algorithms [1, 44]. The VB-method has also been applied to continuous-discrete state-space models, where the state dynamics are represented by a continuous model in time and measurements are discrete [10 – 12].

The Gaussian and variational methods are generally computationally light, but there is no guarantee about the quality of the approximation. Sequential Monte Carlo methods, also called particle filters, are sampling based methods to approximate the filtering distribution and related expectations [22, 47, 49]. The particle filtering methods are computationally more demanding than the Gaussian filtering methods, but under light assumptions, there are theoretical convergence results. The simplest and most widely used particle filtering method is the bootstrap particle filter [25]. More complex strategies for generating the samples use information from the measurements to make the particle filter more efficient [21, 45, 55, 58].

1.2 Research objectives and scope of the research

This thesis collects together research results obtained during my doctoral studies. The published research spans a variety of topics, and one goal of this introduction is to tie them all together by presenting them under a unifying theme of approximate Bayesian inference in stochastic state-space models. The thesis is a compilation thesis and consists of an introductory part (Chapters 1-4) and 4 scientific publications.

Chapters 1-3 give background material and introduction for the research presented in the publications. Chapter 4 provides a conclusion and discussion of the contributions. Finally, the main contributions are presented in the 4 publications attached to the end of the thesis.

Approximation methods can be roughly divided into deterministic and random sampling based methods. The random sampling based methods aim to provide samples from the posterior probability distribution. These methods can be applied for a wide class of models and can approximate the true posterior arbitrarily well, but are computationally heavy. The deterministic methods have typically more restrictions for the mathematical model and do not provide any guarantee about the quality of the approximation, but are computationally light and are often used in real time applications with limited computational resources.

The focus of this research is in applications and development of deterministic approximation methods that improve upon the existing methods and allow the use of more complex mathematical models that better capture the properties of the underlying dynamical system. It is also shown how deterministic methods can be used as part of sampling based methods to make them more efficient and reduce the required computational resources. The main application area of the research is in positioning and navigation, but the results are presented in generic form and applicable to other application areas as well.

The explored topics and contributions are presented in more detail in the following sections.

Using Gaussian filtering methodology for improved statistical GNSS orbit prediction

This topic is explored in publication [PI] and relates to the problem of estimating the initial state of a GNSS satellite using previously received ephemeris data and predicting the state into the future. This has practical applications e.g. in reducing the time required for first positiong fix in an autonomous GNSS positioning device.

The problem is formulated as a Gaussian filtering problem with unknown process noise variance. Major part of the problem is estimating the unknown process noise, that has a significant impact especially to the estimated variance of the predicted orbit. Consistent estimates of the variance are important for assessing the reliability of the prediction results. Finally, the prediction results using the extended Kalman filter are compared to more sophisticated sigma-point based filters.

Developing online algorithms for statistical inference in regression and filtering problems with robust measurement noise models

These algorithms are especially important in applications where the problem needs to be solved in real time with possibly limited computational resources (e.g. online positioning in a mobile phone). Typically a Gaussian noise assumption is used to enable efficient Kalman filter based algorithms for state estimation. Measurement outliers, often encountered in real datasets, however do not fit into the Gaussian assumption and can distort the results.

Publication [PII] presents a variational Bayes based algorithm for linear regression problems with Gaussian scale mixture measurement noise distribution. Special cases from the Gaussian scale mixture family of distributions include the multivariate t-distribution, the multivariate Laplace distribution and the contaminated normal distribution.

Section 3.3 extends the regression results to the nonlinear filtering problem and presents a variational Bayes based outlier robust filtering algorithm for Laplace and contaminated normal measurement noise models. This extends the results presented in [1] and [44] which consider outlier robust filtering based on the multivariate t distribution.

Improving Gaussian filtering based smoothing for continuous-discrete Gaussian models using iterative variational Gaussian algorithm

These results apply for continuous-discrete models, where the transition process is modelled with a continuous stochastic differential equation (SDE) and measurements are taken at discrete time instants. These kind of systems can be found for example in navigation, systems biology and weather forecasting [13, 32, 59]. Computing the exact smoothing solution requires solving the partial differential equations related to the process SDE [27]. The exact solution is tractable only for some special cases and approximations must be used.

In publication [PIII] an iterative variational Bayes based smoothing algorithm is presented for approximating the posterior distribution. The presented variational Gaussian smoothing algorithm extends the results in [10−12] by allowing a more general stochastic process with possibly singular diffusion matrix for describing the system dynamics. Furthermore, a practical algorithm is presented based on the Gaussian filtering based smoothing results in [51]. The presented algorithms use the Gaussian filtering based smoothing results as initial conditions and approximate the

intractable Gaussian integrals with linearisation or sigma-point based methods. It is shown how for some highly nonlinear systems, the variational Gaussian smoother can be used to iteratively improve the Gaussian filtering based smoothing results.

Improving the efficiency of particle Markov chain Monte Carlo methods using twisted particle filters and Gaussian filtering methodology

Particle Markov chain Monte Carlo (PMCMC) methods for model parameter estimation (system identification) use particle filters at each iteration of a MCMC sampler to estimate the marginal likelihood and generate samples from the posterior distribution [5]. The advantage of these methods is that they target the true posterior probability distribution. However, running a particle filter at each iteration means that they are often computationally heavy.

Publication [PIV] shows how the efficiency of the PMCMC algorithm can be improved using twisted particle filters. The twisted particle filter builds upon standard particle filters by adding a special weighting to the sampling law, and can be used to optimally estimate the marginal likelihood of the state-space model [58]. This thesis presents a generalization of the twisted particle that allows a variety of different resampling schemes. Also, a practical implementation of the twisted particle filter is presented for Gaussian models based on the Gaussian filtering methodology. An example with real data shows how the twisted particle filter can deliver computational gains in the particle Markov chain Monte Carlo algorithm compared to standard particle filters.

2 Model description

2.1 Discrete-time state-space model

A general probabilistic discrete time state-space model can be presented in a form

$$\mathbf{x}_0 \sim p(\mathbf{x}_0), \quad \mathbf{x}_k \sim p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}), \quad k \geq 1 \tag{2.1a}$$

$$\mathbf{y}_k \sim p(\mathbf{y}_k \,|\, \mathbf{x}_k), \quad k \geq 0, \tag{2.1b}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state at time t_k, $\mathbf{y}_k \in \mathbb{R}^m$ is the measurement at time t_k, $p(\mathbf{x}_0)$ is the initial distribution, $p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1})$ is the distribution for the state transition process, also called dynamic model, and $p(\mathbf{y}_k \,|\, \mathbf{x}_k)$ is the distribution for the measurement process, also called measurement model.

The system is assumed to be Markovian i.e. the state sequence $\{\mathbf{x}_k\}$ is a Markov sequence and the measurement $\mathbf{y}_k$ given state $\mathbf{x}_k$ is conditionally independent of the state and measurement histories:

$$p(\mathbf{x}_k \,|\, \mathbf{x}_{0:k-1}, \mathbf{y}_{0:k-1}) = p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}), \quad k \geq 1, \tag{2.2a}$$

$$p(\mathbf{y}_k \,|\, \mathbf{x}_{0:k}, \mathbf{y}_{0:k-1}) = p(\mathbf{y}_k \,|\, \mathbf{x}_k), \quad k \geq 0. \tag{2.2b}$$

With these assumptions satisfied, the joint density of the model can be presented by

$$p(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}) = p(\mathbf{x}_0)p(\mathbf{y}_0 \,|\, \mathbf{x}_0) \prod_{s=1}^{k} p(\mathbf{y}_s \,|\, \mathbf{x}_s)p(\mathbf{x}_s \,|\, \mathbf{x}_{s-1}), \quad k \geq 0. \tag{2.3}$$

The following subsections describe the specific state-space models that are considered in this thesis.

2.2 Gaussian model

Gaussian state-space model can be presented in the form

$$\mathbf{x}_0 \sim \mathrm{N}(\mathbf{m}_0^-, \mathrm{P}_0^-), \quad \mathbf{x}_k \sim \mathrm{N}(f_{k-1}(\mathbf{x}_{k-1}), \mathrm{Q}_{k-1}), \quad k \geq 1, \tag{2.4a}$$

$$\mathbf{y}_k \sim \mathrm{N}(h_k(\mathbf{x}_k), \mathrm{R}_k), \quad k \geq 0, \tag{2.4b}$$

where $\mathrm{N}(\mathbf{m}, \mathrm{P})$ denotes a multivariate Gaussian distribution with mean vector $\mathbf{m}$ and covariance matrix P. The transition functions $f_{k-1} \colon \mathbb{R}^n \to \mathbb{R}^n$ and measurement functions $h_k \colon \mathbb{R}^n \to \mathbb{R}^m$ can be any (nonlinear) functions of the state.

An alternative presentation for (2.4), which emphasizes the additive nature of the noise processes, is given by

$$\mathbf{x}_0 \sim \mathrm{N}(\mathbf{m}_0, \mathrm{P}_0), \quad \mathbf{x}_k = f_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}, \quad k \geq 1, \tag{2.5a}$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k, \quad k \geq 0, \tag{2.5b}$$

where $\{\mathbf{w}_k\}$ and $\{\mathbf{v}_k\}$ are zero-mean Gaussian white noise sequences, with covariance matrices $\mathbb{E}[\mathbf{w}_k \mathbf{w}_s^T] = \mathrm{Q}_k \delta_{ks}$ and $\mathbb{E}[\mathbf{v}_k \mathbf{v}_s^T] = \mathrm{R}_k \delta_{ks}$ respectively. In order to satisfy the Markov assumptions (2.2), the random variables $\mathbf{x}_0$, $\{\mathbf{w}_k\}$ and $\{\mathbf{v}_k\}$ are assumed mutually independent.

If the functions f_{k-1} and h_k are linear, the model is called a linear Gaussian model. The linear Gaussian model is an important special case, since it is one of the only special cases that allows an exact solution of the state inference problem.

2.2.1 Models with outlier robust measurement distribution

Real datasets sometimes contain outliers, or extreme observations, that do not fit into the Gaussian assumption about the measurement distribution. Under the Gaussian assumption, even a single outlier can have a significant effect on the results of the statistical inference (see e.g. [24, p. 435]). Outlier-robust inference can be achieved using a measurement distribution with heavier tails than the Gaussian distribution. Here heavy-tailed means distributions that give relatively high probability also for observations far away from the mean.

Fig. 2.1 shows the probability densities and the heavy-tailedness property of two often used outlier robust measurement distributions: the (Student's) t distribution, with degrees of freedom 4, and the Laplace, or double exponential, distribution. Compared to the Gaussian distribution with the same mean and variance, the t and

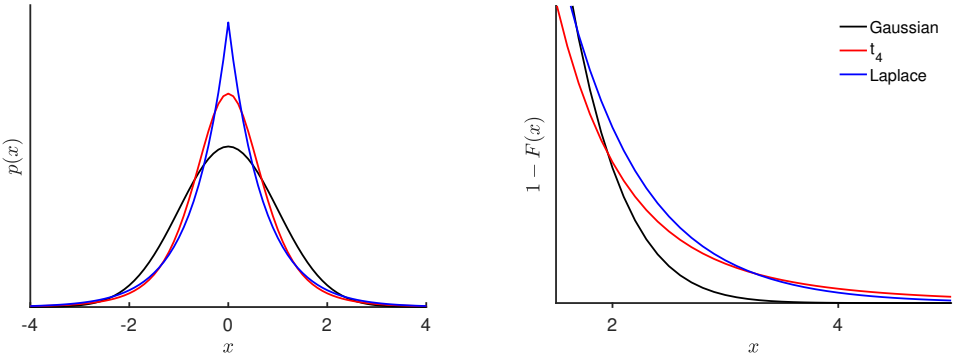


Figure 2.1: Probability density functions (a) and right tail probabilities (b) for Gaussian, t and Laplace distributions with the same mean and variance.

Laplace distributions have more probability mass in the tails of the distribution, and therefore observations far away from the mean are not so rare.

An outlier robust version of the Gaussian model (2.4) is obtained by using a t, Laplace or some other heavy-tailed distribution for the measurement model (2.4b). For computational purposes (see Section 3.3) it is convenient to define the outlier-robust measurement distribution using the hierarchical model

$$\mathbf{y}_k \sim \mathrm{N}(h_k(\mathbf{x}_k), \mathrm{R}_k/u_k), \quad u_k \sim p(u_k), \quad k \geq 0, \tag{2.6}$$

where $\{u_k > 0\}$ is a sequence of independent auxiliary random variables, and the distribution $p(u_k)$ controls the form of the distribution for the measurements. The measurement distribution obtained using (2.6) is also called a Gaussian scale mixture distribution [2], since $p(\mathbf{y}_k \,|\, \mathbf{x}_k)$ is given by

$$p(\mathbf{y}_k \,|\, \mathbf{x}_k) = \int_{\mathbb{R}} \mathrm{N}(\mathbf{y}_k \,|\, h_k(\mathbf{x}_k), \mathrm{R}_k/u_k) p(u_k) \,\mathrm{d}u_k. \tag{2.7}$$

The following list shows how to choose $p(u_k)$ to obtain three common outlier-robust measurement distributions: the multivariate t, multivariate Laplace and multivariate contaminated normal distributions [PII]:

- **The multivariate t distribution** [36], with degrees of freedom ν, is obtained by choosing [39]

$$p(u_k) \propto u_k^{\frac{\nu}{2}-1} e^{-\frac{\nu}{2}u_k}. \tag{2.8}$$

The density (2.7) is given by

$$p(\mathbf{y}_k \,|\, \mathbf{x}_k) \propto \left(1 + \frac{1}{\nu}(\mathbf{y}_k - h_k(\mathbf{x}_k))^T \mathrm{R}_k^{-1}(\mathbf{y}_k - h_k(\mathbf{x}_k))\right)^{-\frac{\nu+m}{2}}. \tag{2.9}$$

- **The multivariate Laplace distribution** [37, p. 296] is obtained by choosing

$$p(u_k) \propto u_k^{-2} e^{-u_k^{-1}}. \tag{2.10}$$

The density (2.7) is given by

$$p(\mathbf{y}_k \,|\, \mathbf{x}_k) \propto \frac{K_{m/2-1}\left(\sqrt{2(\mathbf{y}_k - h_k(\mathbf{x}_k))^T \mathrm{R}_k^{-1}(\mathbf{y}_k - h_k(\mathbf{x}_k))}\right)}{\left(\sqrt{\frac{1}{2}(\mathbf{y}_k - h_k(\mathbf{x}_k))^T \mathrm{R}_k^{-1}(\mathbf{y}_k - h_k(\mathbf{x}_k))}\right)^{m/2-1}}, \tag{2.11}$$

where K_p is a modified Bessel function of the second kind with order p.

- **The multivariate contaminated normal distribution** [39] is obtained by choosing

$$p(u_k) = (1-\epsilon)\delta(u_k - 1) + \epsilon\delta(u_k - 1/c), \tag{2.12}$$

where $0 < \epsilon < 1$ and $c > 1$. The density (2.7) is given by

$$p(\mathbf{y}_k \,|\, \mathbf{x}_k) = (1-\epsilon)\mathrm{N}(\mathbf{y}_k \,|\, h_k(\mathbf{x}_k), \mathrm{R}_k) + \epsilon\mathrm{N}(\mathbf{y}_k \,|\, h_k(\mathbf{x}_k), c\mathrm{R}_k), \tag{2.13}$$

where ϵ gives the probability of obtaining an outlier and c is a variance scaling factor for the outlier observations. The distribution (2.13) is a multivariate generalization of the contaminated normal distribution introduced in [54].

More general Gaussian scale mixture measurement distributions can be obtained by taking the auxiliary variable's distribution $p(u_k)$ from the Gamma or inverse-Gamma families [PII].

2.3 Continuous-discrete Gaussian state-space model

Many real dynamical systems are most easily described in continuous time i.e. using differential equations based on physical principles or empirical models. However, the measurements are nowadays often collected at discrete time instants e.g. using digital sensors or sampling from an analog sensor.

The continuous-discrete Gaussian state-space model is described by

$$\mathbf{x}(t_0) \sim \mathrm{N}(\mathbf{m}_0, \mathrm{P}_0), \quad \dot{\mathbf{x}} = f(\mathbf{x}, t) + \mathrm{L}(t)\mathbf{w}(t), \quad t \geq t_0, \tag{2.14a}$$

$$\mathbf{y}_k \sim \mathrm{N}(h_k(\mathbf{x}(t_k)), \mathrm{R}_k), \quad k \geq 0, \tag{2.14b}$$

where $f\colon \mathbb{R}^{n+1} \to \mathbb{R}^n$ is the transition function, $\mathbf{w}(t) \in \mathbb{R}^d$ is a zero-mean Gaussian white noise stochastic process with covariance function $\mathbb{E}[\mathbf{w}(t)\mathbf{w}^T(\tau)] = \mathrm{Q}(t)\delta(t-\tau)$ and $\mathrm{L}\colon \mathbb{R} \to \mathbb{R}^{n\times d}$. The measurement model is the same as in the discrete-time Gaussian state-space model.

Note that the white noise process is by definition discontinuous everywhere and therefore the differential equation in (2.14a) is not analysable by standard calculus rules. Mathematically rigorous treatment of the continuous-discrete model requires theory of stochastic differential equations and stochastic calculus (see e.g. [27, 43]). However, the presentation (2.14a) is still intuitively useful as a continuous-time version of the discrete-time Gaussian model (2.5a).

The continuous-time model (2.14a) can in theory be converted to the equivalent discrete-time model by solving the integral equation

$$\mathbf{x}(t_k) = \mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} f(\mathbf{x}(t), t)\,\mathrm{d}t + \int_{t_{k-1}}^{t_k} \mathrm{L}(t)\mathbf{w}(t)\,\mathrm{d}t, \tag{2.15}$$

where the latter integral is interpreted as a stochastic Itô integral (see e.g. [27]). For the linear case, with $f(\mathbf{x}, t) = \mathrm{F}(t)\mathbf{x}$, solving (2.15) gives [27, pp. 199-200]

$$\mathbf{x}(t_k) = \Phi(t_k, t_{k-1})\mathbf{x}(t_{k-1}) + \mathbf{w}_{k-1}, \tag{2.16}$$

where the state-transition matrix $\Phi(t_k, t_{k-1})$ is the solution, at time $t = t_k$, of the matrix differential equation

$$\frac{d}{dt}\Phi(t, t_{k-1}) = \mathrm{F}(t)\Phi(t, t_{k-1}), \quad \Phi(t_{k-1}, t_{k-1}) = \mathrm{I}, \tag{2.17}$$

and $\{\mathbf{w}_k\}$ is a zero-mean Gaussian white noise sequence with covariance matrix

$$\mathbb{E}[\mathbf{w}_k \mathbf{w}_s^T] = \left[\int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)\mathrm{L}(\tau)\mathrm{Q}(\tau)\mathrm{L}^T(\tau)\Phi^T(t_{k+1}, \tau)\,\mathrm{d}\tau\right] \delta_{ks}. \tag{2.18}$$

2.4 Example: target tracking

Throughout this introduction, the following example is used to demonstrate the presented methods. The example is motivated by a target tracking problem, where

the goal is to estimate the trajectory of some physical object e.g. a vehicle or a person. The state of the object is given by a vector $\mathbf{x}^T(t) = [\mathbf{r}^T(t), \mathbf{v}^T(t)]$, where $\mathbf{r}(t) \in \mathbb{R}^2$ and $\mathbf{v}(t) \in \mathbb{R}^2$ are the position and velocity of the target in Cartesian coordinates at time $t \geq 0$. A typical model for the state transition process is the constant velocity model given formally by [13, p. 269]

$$\ddot{\mathbf{r}}(t) = \mathbf{w}(t), \tag{2.19}$$

where $\mathbf{w}(t) \in \mathbb{R}^2$ is a zero-mean Gaussian white noise stochastic process with covariance given by

$$\mathbb{E}[\mathbf{w}(t)\mathbf{w}^T(\tau)] = q^2 \mathrm{I}_{2\times 2}\delta(t-\tau), \tag{2.20}$$

where $q > 0$.

The continuous-time transition model is given by

$$\begin{bmatrix} \dot{\mathbf{r}}(t) \\ \dot{\mathbf{v}}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{2\times 2} & \mathrm{I}_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} \end{bmatrix}}_{\mathrm{F}(t)} \begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0_{2\times 2} \\ \mathrm{I}_{2\times 2} \end{bmatrix}}_{\mathrm{L}(t)} \mathbf{w}(t). \tag{2.21}$$

The state transition model is linear, so the continuous-time model can be exactly discretized using (2.16)-(2.18) giving (see also [13, p. 270])

$$\mathbf{x}_k = \mathrm{F}_k \mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \tag{2.22}$$

where

$$\mathrm{F}_k = \begin{bmatrix} \mathrm{I}_{2\times 2} & \mathrm{I}_{2\times 2}\Delta t_k \\ 0_{2\times 2} & \mathrm{I}_{2\times 2} \end{bmatrix}, \tag{2.23}$$

$\Delta t_k = t_k - t_{k-1}$ and $\{\mathbf{w}_k\}_{k\geq 0}$ is a zero-mean Gaussian white noise sequence with covariance

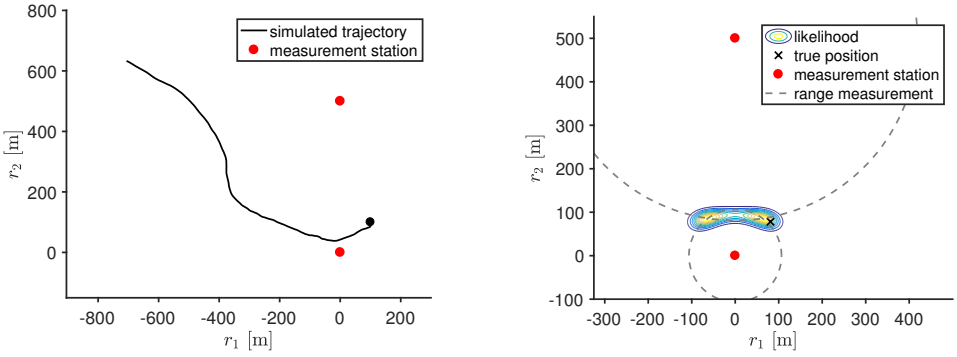
$$\mathbb{E}[\mathbf{w}_k \mathbf{w}_s^T] = q^2 \begin{bmatrix} \frac{(\Delta t_k)^3}{3}\mathrm{I}_{2\times 2} & \frac{(\Delta t_k)^2}{2}\mathrm{I}_{2\times 2} \\ \frac{(\Delta t_k)^2}{2}\mathrm{I}_{2\times 2} & \Delta t_k \mathrm{I}_{2\times 2} \end{bmatrix} \delta_{ks}. \tag{2.24}$$

The measurements come from two stationary rangefinders (measurement stations) located at coordinates $\mathbf{s}_1 = [0, 0]^T$ and $\mathbf{s}_2 = [0, 500]^T$. The measurements are modeled by

$$\mathbf{y}_k = h(\mathbf{r}(t_k), \mathbf{v}(t_k)) + \mathbf{z}_k, \tag{2.25}$$

where

$$h(\mathbf{r}, \mathbf{v}) = \begin{bmatrix} \|\mathbf{r} - \mathbf{s}_1\| \\ \|\mathbf{r} - \mathbf{s}_2\| \end{bmatrix}, \tag{2.26}$$



(a) **(b)**

Figure 2.2: (a) Simulated route and (b) the likelihood at one time instant.

and $\{\mathbf{z}_k\}_{k\geq 1}$ is a zero-mean Gaussian white noise sequence with covariance $\mathbb{E}[\mathbf{z}_k \mathbf{z}_s^T] = \sigma^2 \mathrm{I}_{2\times 2}\delta_{ks}$, $\sigma^2 > 0$. Fig. (2.2) shows a simulated route and the likelihood for a single measurement. Since the measurements do not depend on the velocity $\mathbf{v}$, the likelihood can be plotted by computing $p(\mathbf{y}_k \,|\, \mathbf{r}(t_k))$ on a dense grid for r_1 and r_2. Typically for two range measurements, the likelihood is bimodal, and the modes are located at the intersections of the ring shaped likelihoods of the individual ranges.

3 Methods for sequential Bayesian inference

3.1 Bayesian inference in discrete-time state-space models

Given measurements $\mathbf{y}_{0:t}$, the full solution to the discrete-time state inference problem for the model in Eq. (2.1) is given by the Bayes theorem

$$p(\mathbf{x}_{0:t} \mid \mathbf{y}_{0:t}) = \frac{p(\mathbf{y}_{0:t} \mid \mathbf{x}_{0:t}) p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{0:t})}, \tag{3.1}$$

where $p(\mathbf{y}_{0:t} \mid \mathbf{x}_{0:t})$ is the likelihood, $p(\mathbf{x}_{0:t})$ is the prior and $p(\mathbf{y}_{0:t})$ is a normalization constant, also called marginal likelihood or evidence. Usually, instead of the full posterior distribution, we are interested in specific marginal posterior distributions of the state:

Filtering distribution $p(\mathbf{x}_k \mid \mathbf{y}_{0:k})$ gives the posterior distribution of the state at time k given all the measurements up to that time.

Smoothing distribution $p(\mathbf{x}_k \mid \mathbf{y}_{0:t})$ gives the posterior distribution of the state at time $0 \leq k \leq t$ given all the available measurements.

The filtering distribution can be computed recursively, starting from the prior $p(\mathbf{x}_0)$, using the Bayesian filtering equations

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int_{\mathbb{R}^n} p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) \, \mathrm{d}\mathbf{x}_{k-1}, \quad k \geq 1, \tag{3.2a}$$

$$p(\mathbf{x}_k \mid \mathbf{y}_{0:k}) \propto p(\mathbf{x}_k \mid \mathbf{y}_{0:k-1}) p(\mathbf{y}_k \mid \mathbf{x}_k), \quad k \geq 0. \tag{3.2b}$$

The first equation (3.2a) is often called the prediction step and (3.2b) the update step.

The smoothing distribution can be computed using the backward-time recursion, started from the filtering distribution $p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t})$ at time t, using [34]

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:t}) = p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}) \int_{\mathbb{R}^n} \frac{p(\mathbf{x}_{k+1} \,|\, \mathbf{x}_k) p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:t})}{p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:k})} \, \mathrm{d}\mathbf{x}_{k+1}, \quad t-1 \geq k \geq 0. \tag{3.3}$$

The smoothing presented by (3.3) is also called forward-backward smoothing, since it is based on first computing the forward-time filtering equations and then using the filtering distribution to compute the backward-time smoothing equations. An alternative formulation is the two-filter smoother [18, 23, 35].

In general the integrals in the filtering and smoothing equations are intractable and must be approximated. A special case is the linear Gaussian model, which allows an exact solution that can be computed using the Kalman filter and smoother algorithms. The approximation methods for the non-linear non-Gaussian case can be broadly divided into two classes, deterministic and stochastic approximations. The deterministic approximations replace the posterior distribution with a distribution that allows computationally tractable solution for the filtering and smoothing equations. Typically these are computationally light, but do not give any guarantee about the quality of the approximation. Examples are the Gaussian and factorized variational Bayes approximations.

The stochastic approximation methods are based on generating random samples from the posterior distribution; the samples can be used to approximate the posterior distribution and related expectations with arbitrary accuracy. These methods are however computationally intensive, which limits their use in large-scale or online inference problems. An example is the particle filter based on sequential Monte Carlo sampling.

3.2 Gaussian filtering and smoothing

Gaussian filtering and smoothing is a general framework for solving the Bayesian inference problem in Gaussian state-space models. It is based on approximating the non-Gaussian filtering and smoothing distributions with a Gaussian distribution [26, 49, 50, 60].

3.2.1 Gaussian filter

Assume that we have obtained the Gaussian approximation for the filtering distribution at time t_{k-1}, given by $p(\mathbf{x}_{k-1} \,|\, \mathbf{y}_{0:k-1}) \approx \mathrm{N}(\mathbf{x}_{k-1} \,|\, \mathbf{m}_{k-1}, \mathrm{P}_{k-1})$. Using the Gaussian moment matching [49, pp. 96-97], the prediction step (3.2a) is then given by

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}) \approx \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^-, \mathrm{P}_k^-), \tag{3.4}$$

where

$$\mathbf{m}_k^- = \int_{\mathbb{R}^n} f_{k-1}(\mathbf{x}_{k-1}) \mathrm{N}(\mathbf{x}_{k-1} \,|\, \mathbf{m}_{k-1}, \mathrm{P}_{k-1}) \,\mathrm{d}\mathbf{x}_{k-1}, \tag{3.5a}$$

$$\mathrm{P}_k^- = \int_{\mathbb{R}^n} (f_{k-1}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)(f_{k-1}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T \mathrm{N}(\mathbf{x}_{k-1} \,|\, \mathbf{m}_{k-1}, \mathrm{P}_{k-1}) \,\mathrm{d}\mathbf{x}_{k-1} + \mathrm{Q}_{k-1}. \tag{3.5b}$$

Similarly, using the Gaussian approximation (3.4) and the Gaussian moment matching, the update step (3.2b) is given by

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}) \approx \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k, \mathrm{P}_k), \tag{3.6}$$

where

$$\boldsymbol{\mu}_k = \int_{\mathbb{R}^n} h_k(\mathbf{x}_k) \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^-, \mathrm{P}_k^-) \,\mathrm{d}\mathbf{x}_k, \tag{3.7a}$$

$$\mathrm{S}_k = \int_{\mathbb{R}^n} (h_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)(h_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^-, \mathrm{P}_k^-) \,\mathrm{d}\mathbf{x}_k + \mathrm{R}_k, \tag{3.7b}$$

$$\mathrm{C}_k = \int_{\mathbb{R}^n} (\mathbf{x}_k - \mathbf{m}_k^-)(h_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^-, \mathrm{P}_k^-) \,\mathrm{d}\mathbf{x}_k, \tag{3.7c}$$

$$\mathrm{K}_k = \mathrm{C}_k \mathrm{S}_k^{-1}, \tag{3.7d}$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathrm{K}_k(\mathbf{y}_k - \boldsymbol{\mu}_k), \tag{3.7e}$$

$$\mathrm{P}_k = \mathrm{P}_k^- - \mathrm{K}_k \mathrm{S}_k \mathrm{K}_k^T. \tag{3.7f}$$

The recursion is started with the update step for the prior mean $\mathbf{m}_0^-$ and variance P_0^-. If there is no measurement at $k = 0$, then the update step is left out and $\mathbf{m}_0 = \mathbf{m}_0^-$ and $\mathrm{P}_0 = \mathrm{P}_0^-$.

The resulting Gaussian filter is summarized in Algorithm 1. Figure 3.1 illustrates one step of the Gaussian filter. As can be seen, the Gaussian approximation cannot capture the true shape of the resulting posterior distribution, but is often sufficiently accurate in practical applications.

Algorithm 1 Gaussian filter

1: Set $\mathbf{m}_0$ and P_0 using (3.7)
2: **for** $k = 1$ to t **do**
3: Set $\mathbf{m}_k^-$ and P_k^- using (3.5)
4: Set $\mathbf{m}_k$ and P_k using (3.7)
5: **end for**

If the transition and measurement functions are linear, the Gaussian filter reduces to the Kalman filter [31]. For non-linear transition and measurement functions, the general Gaussian filter is intractable, since the Gaussian integrals in (3.5) and (3.7) cannot be computed explicitly. Many of the common non-linear filters can be seen as special cases of the Gaussian filter when the Gaussian integrals are approximated with a specific numerical integration method. Using first order Taylor series linearization for the transition and measurement functions gives the extended Kalman filter (EKF). Using the cubature or the Gauss-Hermite integration rule to approximate the Gaussian integrals give the cubature Kalman filter (CKF) [6, 60] and Gauss-Hermite Kalman filter (GHKF) [8, 26] respectively. The unscented Kalman filter (UKF) [28, 29, 57] can be also seen as a Gaussian filtering algorithm by a generalization of the CKF integration rule [49, p. 108].

3.2.2 Gaussian smoother

The general Gaussian fixed-interval smoother can be derived in a similar way as the Gaussian filter by assuming that $p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:t}) \approx \mathrm{N}(\mathbf{x}_{k+1} \,|\, \mathbf{m}_{k+1}^s, \mathrm{P}_{k+1}^s)$, and then using the Gaussian moment matching to get the approximation [50], [49, p. 152]

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:t}) \approx \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^s, \mathrm{P}_k^s), \tag{3.8}$$

where

$$\mathrm{D}_{k+1} = \int_{\mathbb{R}^n} (\mathbf{x}_k - \mathbf{m}_k)(f_k(\mathbf{x}_k) - \mathbf{m}_{k+1}^-)^T \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k, \mathrm{P}_k)\, \mathrm{d}\mathbf{x}_k \tag{3.9a}$$

$$\mathrm{G}_k = \mathrm{D}_{k+1}(\mathrm{P}_{k+1}^-)^{-1} \tag{3.9b}$$

$$\mathbf{m}_k^s = \mathbf{m}_k + \mathrm{G}_k(\mathbf{m}_{k+1}^s - \mathbf{m}_{k+1}^-) \tag{3.9c}$$

$$\mathrm{P}_k^s = \mathrm{P}_k + \mathrm{G}_k(\mathrm{P}_{k+1}^s - \mathrm{P}_{k+1}^-)\mathrm{G}_k^T. \tag{3.9d}$$

The resulting Gaussian smoother is summarized in Algorithm 2. Note that the matrices D_{k+1} and G_k depend only on filtering results and can therefore be precomputed and stored during the filtering recursion.

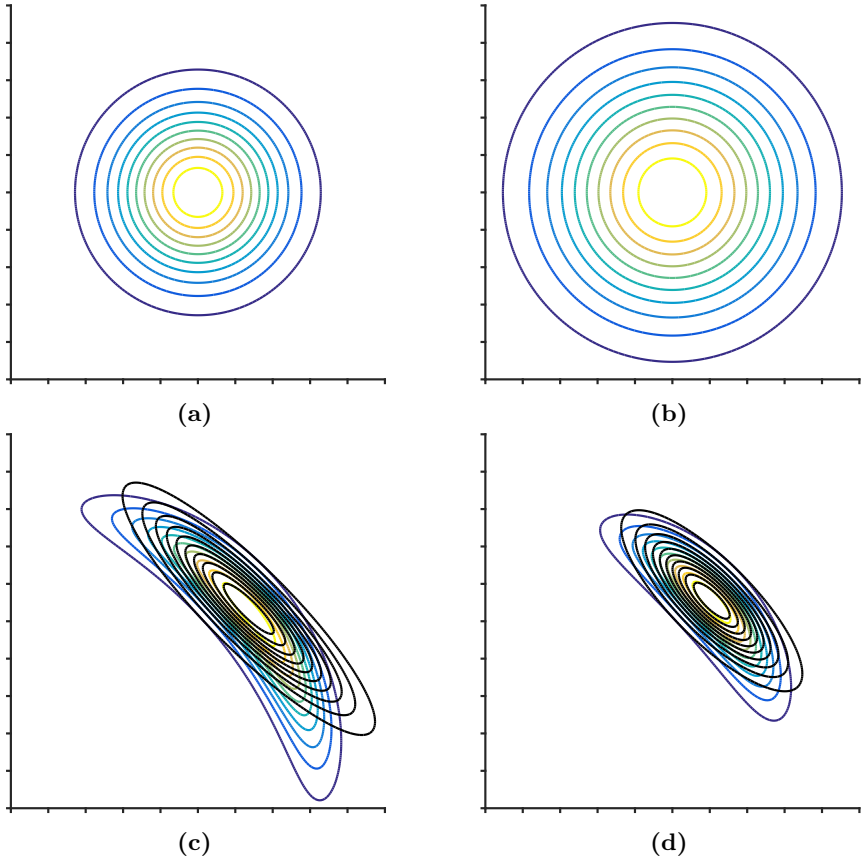


Figure 3.1: Contour lines for the prior distribution (a), prediction distribution (b), measurement likelihood (c) and posterior distribution (d) for the first filtering step of the tracking example. Solid black line in figures (c) and (d) shows contours of the Gaussian approximation computed with EKF.

For a linear model, the Gaussian smoother reduces to the Rauch-Tung-Striebel smoother (RTSS) introduced in [46]. The extended RTSS [20] is obtained by using the first order Taylor series approximation for the transition and measurement functions. The unscented RTSS [48], cubature RTSS [7] and Gauss-Hermite RTSS [50], [49, p. 153-154] can be obtained by using the unscented, cubature and Gauss-Hermite integration rules for the Gaussian integral in (3.9).

Algorithm 2 Gaussian smoother

1: Set $\mathbf{m}_t^s \leftarrow \mathbf{m}_t$ and $\mathrm{P}_t^s \leftarrow \mathrm{P}_t$
2: **for** $k = t-1$ to 0 **do**
3: Set $\mathbf{m}_{k+1}^-$ and P_{k+1}^- using (3.5)
4: Set $\mathbf{m}_k^s$ and P_k^s using (3.9)
5: **end for**

3.2.3 Numerical approximations for the Gaussian integrals

This section provides a brief overview of the Taylor series linearization and sigma-point based numerical approximation methods for the Gaussian integrals in (3.5), (3.7) and (3.9). Detailed discussion of the presented methods can be found e.g. from [49].

Taylor series linearization

The EKF and extended RTSS can be obtained by linearizing f_{k-1} and h_k using the first order Taylor series expansion given by

$$f_{k-1}(\mathbf{x}_{k-1}) \approx f_{k-1}(\mathbf{m}_{k-1}) + \mathrm{F}_{k-1}(\mathbf{x}_{k-1} - \mathbf{m}_{k-1}), \tag{3.10a}$$

$$h_k(\mathbf{x}_k) \approx h_k(\mathbf{m}_k^-) + \mathrm{H}_k(\mathbf{x}_k - \mathbf{m}_k^-), \tag{3.10b}$$

where $\mathrm{F}_{k-1} \in \mathbb{R}^{n \times n}$ is the Jacobian matrix of f_{k-1} evaluated at $\mathbf{m}_{k-1}$, and $\mathrm{H}_k \in \mathbb{R}^{m \times n}$ is the Jacobian matrix of h_k evaluated at $\mathbf{m}_k^-$ given below.

Using (3.10a), the prediction step (3.5) is given by

$$\mathbf{m}_k^- = f_{k-1}(\mathbf{m}_{k-1}) \tag{3.11a}$$

$$\mathrm{P}_k^- = \mathrm{F}_{k-1}\mathrm{P}_{k-1}\mathrm{F}_{k-1}^T + \mathrm{Q}_{k-1}. \tag{3.11b}$$

Using (3.10b), the variables $\boldsymbol{\mu}_k$, S_k and C_k in the update step (3.7) are given by

$$\boldsymbol{\mu}_k = h_k(\mathbf{m}_k^-) \tag{3.12a}$$

$$\mathrm{S}_k = \mathrm{H}_k\mathrm{P}_k^-\mathrm{H}_k^T + \mathrm{R}_k \tag{3.12b}$$

$$\mathrm{C}_k = \mathrm{P}_k^-\mathrm{H}_k^T. \tag{3.12c}$$

And the matrix D_{k+1} in the smoothing step (3.9) is given by

$$\mathrm{D}_{k+1} = \mathrm{P}_k\mathrm{F}_k^T. \tag{3.13}$$

Sigma-point filters and smoothers

The sigma-point approximation is based on approximating the Gaussian integral on a deterministically chosen grid. For a nonlinear function g, the Gaussian integral is approximated using

$$\int g(\mathbf{x})\mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathrm{P})\,\mathrm{d}\mathbf{x} \approx \sum_{i \in I} w^i g(\boldsymbol{\mathcal{X}}^i), \tag{3.14}$$

where the index set I, sigma-points $\{\boldsymbol{\mathcal{X}}^i\}_{i \in I}$ and related weights $\{w^i\}_{i \in I}$ depend on the chosen sigma-point rule. The sigma-points are typically chosen based on the mean and variance of the Gaussian distribution and are given by

$$\boldsymbol{\mathcal{X}}^i = \mathbf{m} + \sqrt{\mathrm{P}}\boldsymbol{\xi}^i, \quad i \in I, \tag{3.15}$$

where the matrix square-root is defined here with $\mathrm{P} = \sqrt{\mathrm{P}}\sqrt{\mathrm{P}}^T$ and the vectors $\boldsymbol{\xi}^i$ depend on the chosen sigma-point rule. Different rules for choosing the sigma-points and weights include e.g. the cubature, unscented and Gauss-Hermite integration rules [49].

The prediction step of the sigma-point filter is given by

$$\boldsymbol{\mathcal{X}}^i_{k-1} = \mathbf{m}_{k-1} + \sqrt{\mathrm{P}_{k-1}}\boldsymbol{\xi}^i, \quad i \in I, \tag{3.16a}$$

$$\mathbf{m}^-_k = \sum_{i \in I} w^i f_{k-1}(\boldsymbol{\mathcal{X}}^i_{k-1}), \tag{3.16b}$$

$$\mathrm{P}^-_k = \sum_{i \in I} w^i (f_{k-1}(\boldsymbol{\mathcal{X}}^i_{k-1}) - \mathbf{m}^-_k)(f_{k-1}(\boldsymbol{\mathcal{X}}^i_{k-1}) - \mathbf{m}^-_k)^T + \mathrm{Q}_{k-1}. \tag{3.16c}$$

The variables $\boldsymbol{\mu}_k$, S_k and C_k in the update step (3.7) are given by

$$\hat{\boldsymbol{\mathcal{X}}}^i_k = \mathbf{m}^-_k + \sqrt{\mathrm{P}^-_k}\boldsymbol{\xi}^i, \quad i \in I, \tag{3.17a}$$

$$\boldsymbol{\mu}_k = \sum_{i \in I} w^i h_k(\hat{\boldsymbol{\mathcal{X}}}^i_k), \tag{3.17b}$$

$$\mathrm{S}_k = \sum_{i \in I} w^i (h_k(\hat{\boldsymbol{\mathcal{X}}}^i_k) - \boldsymbol{\mu}_k)(h_k(\hat{\boldsymbol{\mathcal{X}}}^i_k) - \boldsymbol{\mu}_k)^T + \mathrm{R}_k, \tag{3.17c}$$

$$\mathrm{C}_k = \sum_{i \in I} w^i (\hat{\boldsymbol{\mathcal{X}}}^i_k - \mathbf{m}^-_k)(h_k(\hat{\boldsymbol{\mathcal{X}}}^i_k) - \boldsymbol{\mu}_k)^T. \tag{3.17d}$$

The matrix D_{k+1} in the smoothing step (3.9) is given by

$$\boldsymbol{\mathcal{X}}^i_k = \mathbf{m}_k + \sqrt{\mathrm{P}_k}\boldsymbol{\xi}^i, \quad i \in I, \tag{3.18a}$$

$$\mathrm{D}_{k+1} = \sum_{i \in I} w^i (\boldsymbol{\mathcal{X}}^i_k - \mathbf{m}_k)(f_k(\boldsymbol{\mathcal{X}}^i_k) - \mathbf{m}^-_{k+1})^T. \tag{3.18b}$$

3.3 Outlier-robust Gaussian filtering

In [PII] outlier robust inference is presented for the multivariate regression problem. The regression model can be seen as a special case of a state-space model with the identity function as the transition function, and to better tie the results to the topic of the thesis, this section extends the regression results to the dynamic state-space model case.

Outlier robust filtering and smoothing using the multivariate t distribution is presented in [1, 44]. The results for the robust multivariate regression in [PII] can be combined with the recursive algorithm in [44] to obtain a general recursive outlier-robust filtering and smoothing equations, having the multivariate t, multivariate Laplace and contaminated normal distributions as special cases. For simplicity, the introductory part only deals with the three special cases that are the most useful in practical applications.

The state-space model is given by (2.4a) and (2.6), and the filtering distribution at time k is given by

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}) = \int_{\mathbb{R}} p(\mathbf{x}_k, u_k \,|\, \mathbf{y}_{0:k}) \,\mathrm{d}u_k. \tag{3.19}$$

Computing (3.19) is intractable, and approximate methods must be used.

The variational Bayes (VB) method is a general framework for forming approximate distributions using variational calculus. Usually the VB method refers to finding factorized approximations that minimize the Kullback-Leibler (KL) divergence of the approximation with respect to the original distribution [15, 17, 42].

Applying the VB method for the filtering problem results in a factorized approximation of the form

$$p(\mathbf{x}_k, u_k \,|\, \mathbf{y}_{1:k}) \approx q(\mathbf{x}_k) q(u_k), \tag{3.20}$$

where the distributions $q(\mathbf{x}_k)$ and $q(u_k)$ are chosen to minimize the KL divergence [38] given by

$$\mathrm{KL}\left[q(\mathbf{x}_k)q(u_k) \,||\, p(\mathbf{x}_k, u_k \,|\, \mathbf{y}_{0:k})\right] = \int_{\mathbb{R}} \int_{\mathbb{R}^n} q(\mathbf{x}_k)q(u_k) \log \frac{q(\mathbf{x}_k)q(u_k)}{p(\mathbf{x}_k, u_k \,|\, \mathbf{y}_{0:k})} \,\mathrm{d}\mathbf{x}_k \,\mathrm{d}u_k. \tag{3.21}$$

Note that combining (3.19) and (3.20) gives $p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}) \approx q(\mathbf{x}_k)$.

It can be shown that the optimal distributions $q(\mathbf{x}_k)$ and $q(u_k)$ minimizing (3.21) satisfy [17, pp. 465-466]

$$\log q(\mathbf{x}_k) = \int_{\mathbb{R}} \log p(\mathbf{x}_k, u_k, \mathbf{y}_{0:k}) q(u_k) \, \mathrm{d}u_k + \text{const}, \tag{3.22a}$$

$$\log q(u_k) = \int_{\mathbb{R}^n} \log p(\mathbf{x}_k, u_k, \mathbf{y}_{0:k}) q(\mathbf{x}_k) \, \mathrm{d}\mathbf{x}_k + \text{const}. \tag{3.22b}$$

The equations (3.22) do not give an explicit solution, but can be used as the basis for an iterative coordinate-descent algorithm that converges to the minimum [17, p. 466].

The functional forms for $q(\mathbf{x}_k)$ and $q(u_k)$ are found by computing the integrals in (3.22a) and (3.22b). For (3.22a) we have [44]

$$\begin{aligned} \log q(\mathbf{x}_k) = &-\frac{\bar{u}_k}{2} (\mathbf{y}_k - h_k(\mathbf{x}_k))^T \mathrm{R}_k^{-1} (\mathbf{y}_k - h_k(\mathbf{x}_k)) \\ &- \frac{1}{2} (\mathbf{x}_k - \mathbf{m}_k^-)^T (\mathrm{P}_k^-)^{-1} (\mathbf{x}_k - \mathbf{m}_k^-) + \text{const}, \end{aligned} \tag{3.23}$$

where

$$\bar{u}_k = \int_{\mathbb{R}} u_k q(u_k) \, \mathrm{d}u_k. \tag{3.24}$$

For non-linear measurement function $h_x(\mathbf{x}_k)$ (3.23) does not correspond to any standard distribution, but since it is of the same form as the distribution in the update step of the Gaussian filter, we can use the Gaussian approximation $q(\mathbf{x}_k) \approx \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k, \mathrm{P}_k)$, where the mean $\mathbf{m}_k$ and covariance P_k are computed using

$$\boldsymbol{\mu}_k = \int_{\mathbb{R}^n} h(\mathbf{x}_k) \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^-, \mathrm{P}_k^-) \, \mathrm{d}\mathbf{x}_k, \tag{3.25a}$$

$$\mathrm{S}_k = \int_{\mathbb{R}^n} (h(\mathbf{x}_k) - \mu_k)(h(\mathbf{x}_k) - \mu_k)^T \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^-, \mathrm{P}_k^-) \, \mathrm{d}\mathbf{x}_k + \frac{1}{\bar{u}_k} \mathrm{R}_k, \tag{3.25b}$$

$$\mathrm{C}_k = \int_{\mathbb{R}^n} (\mathbf{x}_k - \mathbf{m}_k^-)(h(\mathbf{x}_k) - \mu_k)^T \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^-, \mathrm{P}_k^-) \, \mathrm{d}\mathbf{x}_k \tag{3.25c}$$

$$\mathrm{K}_k = \mathrm{C}_k \mathrm{S}_k^{-1} \tag{3.25d}$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathrm{K}_k (\mathbf{y}_k - \boldsymbol{\mu}_k) \tag{3.25e}$$

$$\mathrm{P}_k = \mathrm{P}_k^- - \mathrm{K}_k \mathrm{S}_k \mathrm{K}_k^T. \tag{3.25f}$$

For (3.22b) we have

$$\log q(u_k) = -\frac{\bar{\gamma}_k}{2} u_k + \frac{m}{2} \log u_k + \log p(u_k), \tag{3.26}$$

where

$$\bar{\gamma}_k = \operatorname{trace}\left\{\left[\int_{\mathbb{R}^n} (\mathbf{y}_k - h(\mathbf{x}_k))(\mathbf{y}_k - h(\mathbf{x}_k))^T q(\mathbf{x}_k)\,\mathrm{d}\mathbf{x}_k\right] \mathrm{R}_k^{-1}\right\} \tag{3.27}$$

For the three different $p(u_k)$, given in (2.8), (2.10) and (2.12), and corresponding to multivariate t, Laplace and contaminated normal distributions, the distribution $q(u_k)$ in (3.26) is given by the following formulas [PII]

- **Multivariate t distribution**, with ν degrees of freedom: $q(u_k)$ is a Gamma distribution with density

$$q(u_k) \propto u_k^{(\nu+m)/2-1} e^{-u_k(\nu+\bar{\gamma}_k)/2} \tag{3.28}$$

  and mean

$$\bar{u}_k = \frac{\nu + m}{\nu + \bar{\gamma}_k}. \tag{3.29}$$

- **multivariate Laplace distribution**: $q(u_k)$ is a generalized inverse Gaussian distribution with density

$$q(u_k) \propto u_k^{m/2-2} e^{-\frac{1}{2}(\bar{\gamma}_k u_k + 2u_k^{-1})}, \tag{3.30}$$

  and mean

$$\bar{u}_k = \sqrt{\frac{2}{\bar{\gamma}_k}} \frac{K_{m/2}(\sqrt{2\bar{\gamma}_k})}{K_{m/2-1}(\sqrt{2\bar{\gamma}_k})}, \tag{3.31}$$

  where K_p is a modified Bessel function of the second kind with order p.

- **multivariate contaminated normal distribution**: $q(u_k)$ is a two-component mixture with density [1]

$$q(u_k) \propto (1-\epsilon) e^{-\frac{\bar{\gamma}_k}{2}} \delta(u_k - 1) + \epsilon c^{-1} e^{-\frac{\bar{\gamma}_k}{2c}} \delta(u_k - 1/c), \tag{3.32}$$

  and mean

$$\bar{u}_k = \frac{(1-\epsilon) e^{-\bar{\gamma}_k/2} + \epsilon c^{-2} e^{-\bar{\gamma}_k/(2c)}}{(1-\epsilon) e^{-\bar{\gamma}_k/2} + \epsilon c^{-1} e^{-\bar{\gamma}_k/(2c)}}. \tag{3.33}$$

The resulting outlier robust VB filter is summarized in Algorithm 3. Here the parameter $n_{\max}$ determines the number of iterations in the VB update step. Instead of a fixed number of iterations, the termination condition could be obtained by monitoring the variational lower bound [PII]. However, computing the variational lower bound increases the computational cost for each iteration and in practice it is often sufficient to use a fixed number of iterations.

[1] Correction to erroneous formula in [PII] is provided here

Algorithm 3 outlier-robust VB filter

for $i = 1$ to $n_{\max}$ **do**
  Set $\mathbf{m}_0$ and P_0 using (3.25)
  $\Sigma_0 \leftarrow \int (\mathbf{y}_0 - h_0(\mathbf{x}_0))(\mathbf{y}_0 - h_0(\mathbf{x}_0))^T \mathrm{N}(\mathbf{x}_0 \,|\, \mathbf{m}_0^-, \mathrm{P}_0^-)\, \mathrm{d}\mathbf{x}_0$
  $\bar{\gamma}_0 \leftarrow \mathrm{trace}(\Sigma_0 \mathrm{R}_0^{-1})$
  Set $\bar{u}_k$ using (3.29) for multivariate t, (3.31) for multivariate Laplace, (3.33) for multivariate contaminated normal
end for
for $k = 1$ to t **do**
  $\mathbf{m}_k^- \leftarrow \int f_{k-1}(\mathbf{x}_{k-1}) \mathrm{N}(\mathbf{x}_{k-1} \,|\, \mathbf{m}_{k-1}, \mathrm{P}_{k-1})\, \mathrm{d}\mathbf{x}_k$
  $\mathrm{P}_k^- \leftarrow \int (f_{k-1}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)(f_{k-1}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T \mathrm{N}(\mathbf{x}_{k-1} \,|\, \mathbf{m}_{k-1}, \mathrm{P}_{k-1})\, \mathrm{d}\mathbf{x}_k + \mathrm{Q}_{k-1}$
  $\bar{u}_k \leftarrow 1$
  for $i = 1$ to $n_{\max}$ **do**
    Set $\mathbf{m}_k$ and P_k using (3.25)
    $\Sigma_k \leftarrow \int (\mathbf{y}_k - h_k(\mathbf{x}_k))(\mathbf{y}_k - h_k(\mathbf{x}_k))^T \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k, \mathrm{P}_k)\, \mathrm{d}\mathbf{x}_k$
    $\bar{\gamma}_k \leftarrow \mathrm{trace}(\Sigma_k \mathrm{R}_k^{-1})$
    Set $\bar{u}_k$ using (3.29) for multivariate t, (3.31) for multivariate Laplace, (3.33) for multivariate contaminated normal
  end for
end for

3.4 Particle filtering

The particle filter, or sequential Monte Carlo method, uses sequential importance sampling to draw samples from the filtering distribution. The set of samples, or particles, can then be used to approximate the filtering distribution and related expectations. The particle filter can be used for the general state-space model (2.1).

Let $\{\mathbf{x}_k^i \in \mathbb{R}^{d_x}\}_{i=1}^n$ denote the set of particles and $\{w_k^i \in [0,1]\}_{i=1}^n$ the set of related weights at time k. The weights satisfy $\sum_{i=1}^n w_k^i = 1$. The approximation for the filtering distribution is given by

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i). \tag{3.34}$$

The particles $\mathbf{x}_k^i$ are drawn from an importance distribution $q(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}^i, \mathbf{y}_{0:k})$ and the weights are evaluated using

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{y}_k \,|\, \mathbf{x}_k^i) p(\mathbf{x}_k^i \,|\, \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i \,|\, \mathbf{x}_{k-1}^i, \mathbf{y}_{0:k})}. \tag{3.35}$$

To avoid problems with degeneracy, the particle filters use a resampling step to discard particles with small weights and duplicate particles with large weights [47].

The particle filter algorithm is presented in Algorithm 4. Here resampling is done at step i if the effective sample size, given by

$$N_{\text{eff}} = \sum_{i=1}^{N} \frac{1}{(w_k^i)^2}, \tag{3.36}$$

goes below a predefined threshold N_{thr}.

Using Equation (3.34) the expectations with respect to the filtering distribution can be approximated using

$$\mathrm{E}[g(\mathbf{x}_k) \,|\, \mathbf{y}_{0:k}] \approx \sum_{i=1}^{N} w_k^i g(\mathbf{x}_k^i). \tag{3.37}$$

As the number of particles increases, this approximation approaches the true value of the expectation by the law of large numbers [19].

Taking the transition distribution as the importance distribution, i.e. $q(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \mathbf{y}_{0:k}) = f(\mathbf{x}_k \,|\, \mathbf{x}_{k-1})$, gives the bootstrap particle filter [25]. The bootstrap particle filter is simple to implement, but can be inefficient since the particles are propagated without using any measurement information. An optimal importance distribution, in the sense of minimizing the variance of the weights in a single time step, can be shown to be $q(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \mathbf{y}_{0:k}) \propto p(\mathbf{y}_k \,|\, \mathbf{x}_k) p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1})$ [21]. For nonlinear and non-Gaussian measurement models the optimal importance distribution is often intractable, but can be approximated e.g. using Kalman filter extensions [21, 55]. The auxiliary particle filter [45] uses the measurement information in the resampling step and effectively mimics the use of the optimal importance distribution.

3.5 Filtering and smoothing in continuous-discrete models

For the continuous-discrete model, the state $\mathbf{x}(t)$ is now continuous in time. Therefore the filtering distribution using measurements up to time k is defined for the whole time interval $[t_k, t_{k+1})$ and denoted by

$$p(\mathbf{x}(t) \,|\, \mathbf{y}_{0:k}), \quad t_k \leq t < t_{k+1}. \tag{3.38}$$

Similarly, the smoothing distribution is defined for time interval $[t_0, t_t]$ and denoted by

$$p(\mathbf{x}(t) \,|\, \mathbf{y}_{0:t}), \quad t_0 \leq t \leq t_t. \tag{3.39}$$

For a fully continuous (i.e. both dynamics and measurements are continuous in time) linear Gaussian model, the solution to the filtering problem is given by the

Algorithm 4 Particle filter

Sample $\mathbf{x}_0^i \sim p_0(\mathbf{x}_0)$ and set $w_0^i \leftarrow p(\mathbf{y}_0 \mid \mathbf{x}_0^i)$ for $1 \leq i \leq N$
for $1 \leq k \leq t$ **do**
  for $1 \leq i \leq N$ **do**
    Sample $\mathbf{x}_k^i \sim q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i)$
    Set $w_k^i \leftarrow w_{k-1}^i \frac{p(\mathbf{y}_k \mid \mathbf{x}_k^i) p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}$
  end for
  Set $N_{\text{eff}} \leftarrow \sum_{i=1}^{N} \frac{1}{(w_k^i)^2}$
  if $N_{\text{eff}} < N_{\text{thr}}$ **then**
    Set $(\mathbf{x}_k^i)_{i=1}^N \leftarrow \text{resample}((\mathbf{x}_k^i)_{i=1}^N, (w_k^i)_{i=1}^N)$
    Set $w_k^i \leftarrow 1/N$ for all i
  end if
end for

Kalman-Bucy filter [30]. The continuous-discrete Kalman filter is a combination of the continuous-time Kalman-Bucy filter and the discrete-time Kalman filter, where the Kalman-Bucy prediction equations are used between observations and the Kalman filter update step at observation times [27]. The continuous-time Rauch-Tung-Striebel smoother [46] uses the filtering solution and gives a recursive solution for the smoothing problem in the linear continuous-discrete system.

3.5.1 Gaussian filtering and smoothing for the continuous-discrete model

For non-linear models, the filtering and smoothing distributions are no longer Gaussian, but similarly to the discrete-time case, we can consider Gaussian approximations for these densities. This means approximating the filtering and smoothing densities with

$$p(\mathbf{x}(t) \mid \mathbf{y}_{0:k}) \approx \mathrm{N}(\mathbf{x}(t) \mid \mathbf{m}(t), \mathrm{P}(t)), \quad t_k \leq t < t_{k+1},\ 0 \leq k \leq t, \tag{3.40a}$$

$$p(\mathbf{x}(t) \mid \mathbf{y}_{0:k}) \approx \mathrm{N}(\mathbf{x}(t) \mid \mathbf{m}^s(t), \mathrm{P}^s(t)), \quad t_0 \leq t \leq t_t, \tag{3.40b}$$

where $\mathbf{m}_k(t)$ and $\mathbf{m}^s(t)$ are the mean functions and $\mathrm{P}(t)$ and $\mathrm{P}^s(t)$ are the auto-covariance functions for the approximating Gaussian distributions. The filtering and smoothing problem now reduces to finding the expressions for the mean and covariance functions.

In this thesis, two methods for computing the Gaussian approximations are presented. The classical Gaussian approximation is based on recursions similar to the discrete-

time case, and gives approximate solutions for the mean and covariance functions in the filtering and smoothing problem [51]. Another solution is based on the variational Bayes methodology, where the mean and covariance functions for the smoothing solution are chosen to minimize the KL-divergence between the true and approximate solutions [11, 12]. In some sense, this can be seen as an iterative way to improve the classical Gaussian approximation [PIII].

3.5.2 Gaussian filtering based approximation

In the prediction step, the mean and covariance functions are propagated from time t_{k-1} to time t_k by solving the differential equations

$$\dot{\mathbf{m}}(t) = \int_{\mathbb{R}^n} f(\mathbf{x}, t)\mathrm{N}(\mathbf{x} \,|\, \mathbf{m}(t), \mathrm{P}(t))\,\mathrm{d}\mathbf{x}, \tag{3.41a}$$

$$\dot{\mathrm{P}}(t) = \int_{\mathbb{R}^n} \left[(\mathbf{x} - \mathbf{m}(t))f^T(\mathbf{x}, t) + f(\mathbf{x}, t)(\mathbf{x} - \mathbf{m}(t))^T\right] \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}(t), \mathrm{P}(t))\,\mathrm{d}\mathbf{x} + \Sigma(t), \tag{3.41b}$$

with initial conditions $\mathbf{m}_{k-1}(t_{k-1})$ and $\mathrm{P}(t_{k-1})$ obtained from the previous step.

In the update step, the information from the latest measurement $\mathbf{y}_k$ is used to update the predicted estimates $\mathbf{m}(t_k^-)$ and $\mathrm{P}(t_k^-)$ using equations

$$\boldsymbol{\mu}_k = \mathbb{E}[h_k(\mathbf{x}(t_k^-)), \tag{3.42a}$$

$$\mathrm{S}_k = \mathbb{E}\left[(h_k(\mathbf{x}(t_k^-)) - \boldsymbol{\mu}_k)(h_k(\mathbf{x}(t_k^-)) - \boldsymbol{\mu}_k)^T\right] + \mathrm{R}_k, \tag{3.42b}$$

$$\mathrm{K}_k = \mathbb{E}\left[(\mathbf{x}(t_k^-) - \mathbf{m}(t_k^-))(h_k(\mathbf{x}(t_k^-)) - \boldsymbol{\mu}_k)^T\right] \mathrm{S}_k^{-1}, \tag{3.42c}$$

$$\mathbf{m}(t_k) = \mathbf{m}(t_k^-) + \mathrm{K}_k\left(\mathbf{y}_k - \boldsymbol{\mu}_k\right), \tag{3.42d}$$

$$\mathrm{P}(t_k) = \mathrm{P}(t_k^-) - \mathrm{K}_k\mathrm{S}_k\mathrm{K}_k^T. \tag{3.42e}$$

Gaussian approximation for the smoothing distribution can be derived using several different approaches. A numerically stable and computationally light solution is given by [51]

$$\dot{\mathbf{m}}^s(t) = \mathbb{E}[f(\mathbf{x}(t), t)] + \left[\mathbb{E}[f(\mathbf{x}(t), t)(\mathbf{x}(t) - \mathbf{m}(t))^T] + \Sigma(t)\right] \mathrm{P}^{-1}(t)(\mathbf{m}^s(t) - \mathbf{m}(t)), \tag{3.43a}$$

$$\begin{aligned}\dot{\mathrm{P}}^s(t) = {} & (\mathbb{E}[f(\mathbf{x}(t), t)(\mathbf{x}(t) - \mathbf{m}(t))^T] + \Sigma(t))\mathrm{P}^{-1}(t)\mathrm{P}^s(t) \\ & + \mathrm{P}^s(t)\mathrm{P}^{-1}(t)(\mathbb{E}[f(\mathbf{x}(t), t)(\mathbf{x}(t) - \mathbf{m}(t))^T]^T + \Sigma(t)) - \Sigma(t),\end{aligned} \tag{3.43b}$$

where the expectations are with respect to the filtering distribution.

If the Jacobian $\mathrm{F}_{\mathbf{x}}(\mathbf{x}, t)$ of $f(\mathbf{x}, t)$ is available, we can use the result

$$\mathbb{E}[f(\mathbf{x}, t)(\mathbf{x} - \mathbf{m})^T] = \mathbb{E}[\mathrm{F}_{\mathbf{x}}(\mathbf{x}, t)]\mathrm{P} \tag{3.44}$$

in the filtering and smoothing equations. Since the expectations in the smoothing equations are with respect to the filtering density, they can be precomputed already during the filtering stage.

3.5.3 Variational Gaussian approximation

The variational Gaussian approximation, first presented in [11], for the continuous-discrete smoothing problem is based on approximating the intractable smoothing process with a linear stochastic differential equation

$$d\mathbf{x} = [-\mathrm{A}(t)\mathbf{x}(t) + \mathbf{b}(t)]\, dt + \sqrt{\Sigma(t)}\, d\boldsymbol{\beta}(\boldsymbol{t}), \tag{3.45}$$

where $\mathrm{A}(t)$ and $\mathbf{b}(t)$ are parameters of the approximation and $\boldsymbol{\beta}(\boldsymbol{t})$ is a Brownian stochastic process with identity diffusion matrix. The matrix $\Sigma(t)$ is called the effective diffusion matrix of the original stochastic differential equation in Eq. (2.14a) and is given by

$$\Sigma(t) = \mathrm{L}(t)\mathrm{Q}(t)\mathrm{L}^T(t). \tag{3.46}$$

For simplicity, only the case of non-singular effective diffusion matrix is considered in this introduction. Derivation of the equations for some singular cases can be found in [PIII].

The solution to the linear stochastic differential equation (3.45) is a Gaussian process, with mean and variance defined by ordinary differential equations

$$\dot{\mathbf{m}}(t) = -\mathrm{A}(t)\mathbf{m}(t) + \mathbf{b}(t), \tag{3.47a}$$

$$\dot{\mathrm{P}}(t) = -\mathrm{A}(t)\mathrm{P}(t) - \mathrm{P}(t)\mathrm{A}^T(t) + \Sigma(t). \tag{3.47b}$$

The parameters $\mathrm{A}(t)$ and $\mathbf{b}(t)$ are found by minimizing the KL-divergence of the approximating Gaussian measure and the true smoothing measure. The optimal solution can be shown to satisfy the Euler-Lagrange equations given by [11, 12]

$$\dot{\boldsymbol{\lambda}}(t) = \mathrm{A}^T(t)\boldsymbol{\lambda}(t) - \nabla_{\mathbf{m}}\mathbb{E}[e(\mathbf{x}, t)], \tag{3.48a}$$

$$\dot{\Psi}(t) = \Psi(t)\mathrm{A}(t) + \mathrm{A}^T(t)\Psi(t) - \nabla_{\mathrm{P}}\mathbb{E}[e(\mathbf{x}, t)], \tag{3.48b}$$

$$\mathrm{A}(t) = -\mathbb{E}[\nabla_{\mathbf{x}} f(\mathbf{x}, t)]^T + 2\Sigma(t)\Psi(t), \tag{3.48c}$$

$$\mathbf{b}(t) = \mathbb{E}[f(\mathbf{x}, t)] + \mathrm{A}(t)\mathbf{m}(t) - \Sigma(t)\boldsymbol{\lambda}(t), \tag{3.48d}$$

where $\boldsymbol{\lambda}(t)$ and $\Psi(t)$ are auxiliary Lagrange functions and

$$e(\mathbf{x},t) = \frac{1}{2}[f(\mathbf{x},t) + \mathrm{A}(t)\mathbf{x} - \mathbf{b}(t)]^T \Sigma^{-1}(t)[f(\mathbf{x},t) + \mathrm{A}(t)\mathbf{x} - \mathbf{b}(t)]. \tag{3.49}$$

For notational convenience, the time dependence of $\mathbf{x}$ is suppressed and we write $\mathbf{x} = \mathbf{x}(t)$.

The information from the observations is taken into account by boundary conditions for the Lagrange functions at times t_k, given by

$$\boldsymbol{\lambda}(t_k^+) = \boldsymbol{\lambda}(t_k^-) + \nabla_{\mathbf{m}} \mathbb{E}[u_k(\mathbf{x})], \tag{3.50}$$

$$\Psi(t_k^+) = \Psi(t_k^-) + \nabla_{\mathrm{P}} \mathbb{E}[u_k(\mathbf{x})], \tag{3.51}$$

where

$$u_k(\mathbf{x}) = \frac{1}{2}\left[\mathbf{y}_k - h_k(\mathbf{x})\right]^T \mathrm{R}_k^{-1}\left[\mathbf{y}_k - h_k(\mathbf{x})\right]. \tag{3.52}$$

The expectations in the Euler-Lagrange equations are with respect to the approximating Gaussian smoothing distribution.

Given initial estimates $\mathrm{A}_0(t)$ and $\mathbf{b}_0(t)$, the Euler-Lagrange equations can be solved using an iterative algorithm, where $\mathbf{m}_{i+1}(t)$ and $\mathrm{P}_{i+1}(t)$ are first computed by solving the differential equations (3.47) forward in time. Next, new estimates $\Psi_{i+1}(t)$ and $\boldsymbol{\lambda}_{i+1}(t)$ for the Lagrange functions are computed by solving the differential equations (3.48a) and (3.48b) backward in time. Finally, new estimates $\mathrm{A}_{i+1}(t)$ and $\mathbf{b}_{i+1}(t)$ are obtained by computing (3.48c) and (3.48d).

The iteration is sensitive to the initial estimates $\mathrm{A}_0(t)$ and $\mathbf{b}_0(t)$. In [PIII] we suggest to use the classical Gaussian smoothing solution as the starting point for the iteration. This requires finding $\mathrm{A}(t)$ and $\mathbf{b}(t)$ corresponding to Eqs. (3.43), which are given by [PIII]

$$\mathrm{A}_0(t) = -\mathbb{E}[\mathrm{F}_{\mathbf{x}}(\mathbf{x},t)] - \Sigma(t)(\mathrm{P}^f(t))^{-1} - (\mathrm{P}^s(t))^{-1}) \tag{3.53a}$$

$$\mathbf{b}_0(t) = \mathbb{E}[f(\mathbf{x},t)] - \mathbb{E}[\mathrm{F}_{\mathbf{x}}(\mathbf{x},t)]\mathbf{m}^f(t) - \Sigma(t)[(\mathrm{P}^f(t))^{-1}\mathbf{m}^f(t) - (\mathrm{P}^s(t))^{-1}\mathbf{m}^s(t)]. \tag{3.53b}$$

The expectations are with respect to the Gaussian filtering distribution, which is denoted here by $\mathrm{N}(\mathbf{m}^f(t), \mathrm{P}^f(t))$ to differentiate the mean and variance from the variational solution.

The resulting VB smoothing algorithm is summarized in Algorithm 5.

In a practical implementation of the variational Gaussian smoother, the values of the continuous functions are computed and stored at discrete time points. The

Algorithm 5 VB smoother

1: Compute $\mathbf{m}^f(t)$, $\mathbf{m}^s(t)$, $\mathrm{P}^f(t)$ and $\mathrm{P}^s(t)$ using the Gaussian filtering and smoothing Eqs. (3.41), (3.42) and (3.43)
2: Compute $\mathrm{A}_0(t)$ and $\mathbf{b}_0(t)$ using Eqs. (3.53)
3: **for** $1 \leq i \leq M$ **do**
4: Compute $\mathbf{m}_i(t)$ and $\mathrm{P}_i(t)$ using Eqs. (3.47)
5: Compute $\boldsymbol{\lambda}_i(t)$ and $\Psi_i(t)$ using Eqs. (3.48a), (3.48b), (3.50) and (3.51)
6: Compute $\mathrm{A}_i(t)$ and $\mathbf{b}_i(t)$ using Eqs. (3.48c) and (3.48d)
7: **end for**

intractable expectations can be approximated using similar methods as the Gaussian filters i.e. Taylor series based linearization or sigma-point approximations [PIII]. The differential equations can be solved using any standard numerical solver such as Euler or Runge–Kutta methods.

3.6 Example: state estimation for the target tracking problem

Here the presented methods are demonstrated using simulated data from the target tracking problem presented in Section 2.4. The noise parameters are set to $q = 0.1$ and $\sigma = 10$. The time step is fixed to $\Delta t_k = 1$. One simulated path contains 1000 samples and is started from initial conditions $\mathbf{x}(0) = [100, 100, 0, 0]^T$. The prior distribution is chosen to be Gaussian with covariance matrix

$$\mathrm{P}^-(0) = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 10^2 & 0 & 0 \\ 0 & 0 & 0.01^2 & 0 \\ 0 & 0 & 0 & 0.01^2 \end{bmatrix} \tag{3.54}$$

and mean $\mathbf{m}^-(0) \sim \mathrm{N}(\mathbf{x}(0), \mathrm{P}^-(0))$ that is sampled separately for each simulation. These parameters model a situation where we have a relatively good approximation for the initial position and the target is assumed to start from a standstill.

3.6.1 Gaussian filtering

Consider first solving the target tracking problem using the Gaussian filtering methodology. The dynamic model is linear in the state, so the prediction step in

Table 3.1: Average root mean square error (RMSE) computed over 100 simulated data sets.

Filtering method	RMSE
EKF	5.1874
CKF	4.6758
PF	4.4582

the Gaussian filter reduces to the standard Kalman filter prediction step given by

$$\begin{aligned} \mathbf{m}_k^- &= \mathrm{F}\mathbf{m}_{k-1} \\ \mathrm{P}_k^- &= \mathrm{F}\mathrm{P}_{k-1}\mathrm{F}^T + \mathrm{Q}, \end{aligned}$$

where F and Q are given by Eqs. (2.23) and (2.24) respectively. For the update step, we need to solve the Gaussian expectations over the non-linear measurement function. In this example, the extended Kalman filter and cubature Kalman filter methods are used. Figure (3.2) shows mean estimates of the simulated path with the EKF and CKF methods. Note that the EKF and CKF paths are very close to each other. As a reference, the figure also shows an estimated path computed using a standard bootstrap particle filter with 10^4 particles.

To compare the average performance of the methods, Table 3.1 shows average root mean square errors computed over 100 simulations. For each simulation the initial mean $m^-(0)$ was randomly sampled from $\mathrm{N}(\mathbf{x}(0), \mathrm{P}^-(0))$ with $\mathbf{x}(0) = [100, 100, 0, 0]^T$ and $\mathrm{P}^-(0)$ given in Eq. (3.54). As expected, the EKF produces on average the worst accuracy and the particle filter is the most accurate. However, differences with the filters are observed to be quite small in this example. One reason is that the nonlinearities in this example are not that severe and there is not that much benefit in using more accurate methods than the Taylor series based linearisation. Bigger differences between the filters are expected in highly nonlinear problems, see e.g. the re-entry example given in [28].

It should be also noted that the sigma-point based filters do not require any derivatives of the process or measurement functions. This can be a benefit in problems where evaluation of the Jacobian matrix is difficult.

3.6.2 Outlier robust filtering

To demonstrate the effect of outliers, consider a case in which the range measurements are corrupted by outliers. This is simulated by sampling the measurements

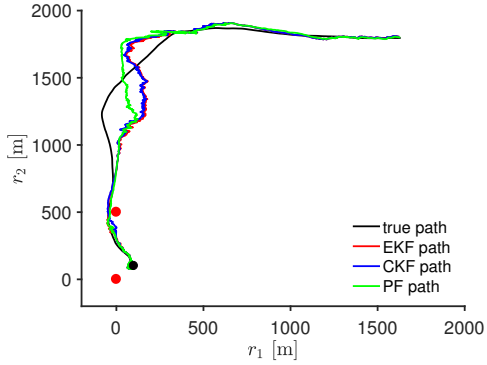


Figure 3.2: Mean estimates estimates of path computed using EKF, CKF and particle filter with $N = 10000$ particles.

from a Gaussian mixture (i.e. contaminated normal)

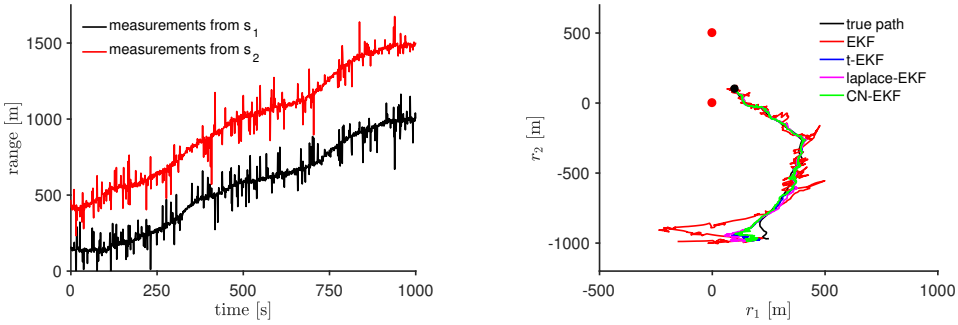
$$\mathbf{y}_k \sim (1-\epsilon)\mathrm{N}(\mathbf{y}_k \,|\, h(\mathbf{x}_k), \mathrm{R}) + \epsilon \mathrm{N}(\mathbf{y}_k \,|\, h(\mathbf{x}_k), 100\mathrm{R}), \tag{3.55}$$

where the outlier probability is set to $\epsilon = 0.2$. That is, the outlying observations have the same mean but 100 times larger variance than "clean" measurements.

Figure (3.3a) shows simulated range measurements from the contaminated normal distribution for one simulated path. The outliers clearly stand out as spikes in the range measurements. Figure (3.3b) shows estimates for one simulated path using EKF and different VB filters with t distribution with 4 degrees of freedom, Laplace distribution and contaminated normal distribution, with parameters set to match the mean and variance of the actual measurement distribution. It can be seen that the EKF has large deviations from the true path, caused by the outlying observations. All the tested VB filters are seen to follow the true path much better and to be less affected by the outliers.

Table 3.2 shows the average root mean square errors over 100 simulated data sets. These results also show that the EKF with Gaussian measurement distribution is heavily influenced by the outlier observations and gives worse RMSE results compared to the robust alternatives. Note that in this example, the contaminated normal model produced the best results, which is expected as it models exactly the simulated measurement noise process.

In practice, if no knowledge about the true measurement distribution is available, the choice of best outlier robust filter is often not clear. Computationally the t and contaminated normal distribution based filters are more convenient than the Laplace



(a) (b)

Figure 3.3: (a) Range measurements simulated from the contaminated normal distribution with outlier probability $\epsilon = 0.2$ and covariance scale $c = 100$. (b) Mean estimates of path computed using EKF and outlier robust VB filters using multivariate t distribution with degrees of freedom $\nu = 4$, Laplace distribution and contaminated normal distribution with outlier probability $\epsilon = 0.2$ and covariance scale $c = 100$.

Table 3.2: Average root mean square error (RMSE) computed over 100 simulated data sets.

Filtering method	RMSE
EKF	11.0151
VB t	4.1885
VB Laplace	3.9089
VB CN	3.0293

distribution based filter that requires the evaluation of a modified Bessel function in the update step. The t and contaminated normal distribution based filters have additional parameters, which provide more flexibility, but can be difficult to choose in practice. Especially the choice of outlier probability and outlier covariance scale for the contaminated normal model might be difficult if the true outlier process is not known. A method for estimating the parameters from data in the linear regression problem is proposed in [PII]. Extending this method to the nonlinear filtering problem is not straightforward and is beyond the scope of this introduction.

3.6.3 Continuous-discrete filtering and smoothing

This section demonstrates solving the continuous-discrete smoothing problem using the presented Gaussian and VB methods. The dynamic model is linear, so the

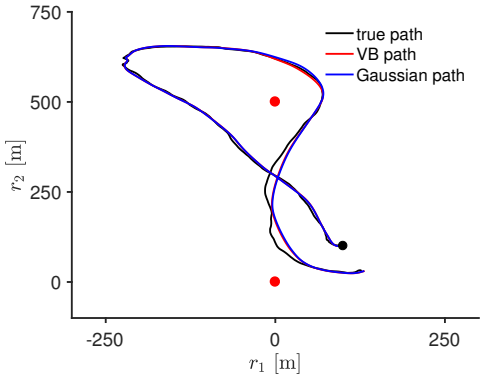


Figure 3.4: Mean estimates estimates of path computed using Gaussian and VB smoothing methods.

Table 3.3: Average root mean square error (RMSE) computed over 100 simulated data sets.

Smoothing method	RMSE
Gaussian	3.0130
VB	3.0456

Gaussian continuous-discrete filtering prediction step is given by

$$\dot{\mathbf{m}}^f(t) = \mathrm{F}\mathbf{m}^f(t) \tag{3.56}$$

$$\dot{\mathrm{P}}^f(t) = \mathrm{F}\mathrm{P}^f(t) + \mathrm{P}^f(t)\mathrm{F}^T + \Sigma. \tag{3.57}$$

For this example, only the Taylor series linearization method for computing the Gaussian expectations is considered. This leads to the same update step as in the EKF at measurement times. The differential equations for the smoothed mean and variance are given by

$$\dot{\mathbf{m}}^s(t) = \mathrm{F}\mathbf{m}^s(t) + \Sigma(\mathrm{P}^f(t))^{-1}(\mathbf{m}^s(t) - \mathbf{m}^f(t)), \tag{3.58}$$

$$\dot{\mathrm{P}}^s(t) = \mathrm{F}\mathrm{P}^s(t) + \mathrm{P}^s(t)\mathrm{F}^T + \Sigma(\mathrm{P}^f(t))^{-1}\mathrm{P}^s(t) + \mathrm{P}^s(t)(\mathrm{P}^f(t))^{-1}\Sigma - \Sigma. \tag{3.59}$$

The Gaussian filtering and smoothing solution is used as initial conditions for the VB smoother using Eq. (3.53). Details for computing the expectations in the VB smoother equations (3.48) using Taylor series linearization can be found in [PIII].

Figure 3.4 shows mean estimates for a simulated dataset and Table 3.3 shows the RMSE computes over 100 datasets. For this example, the Gaussian and VB solutions are very close and no clear difference can be seen.

4 Parameter inference in state-space models

4.1 Parameter inference

Previous chapter presented methods for estimating the unknown state of a dynamic system given a known model for the system dynamics and measurements. However, often in practice the exact model is not known and must be estimated. In this thesis, is consider the case where the functional form of the model is fixed, but might include some unknown parameters e.g. the measurements are modelled as normally distributed with unknown variance. Moreover, the parameters are static i.e. the parameter values are not expected to change during the time interval of interest.

Let $\boldsymbol{\theta}$ denote the unknown parameters in the state-space model. These can include for example the measurement and process noise variances, and unknown parameters in the functions describing the process dynamics and measurements.

In the Bayesian framework, the parameters are considered as random variables and estimates are computed using the posterior distribution

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}) \propto p_{\boldsymbol{\theta}}(\mathbf{y}_{0:t}) p(\boldsymbol{\theta}), \tag{4.1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$, $p_{\boldsymbol{\theta}}(\mathbf{y}_{0:t})$ is the likelihood and $p(\boldsymbol{\theta})$ is the prior.

The likelihood term can be evaluated recursively for $k \geq 1$ with

$$p(\mathbf{y}_{0:k} \,|\, \boldsymbol{\theta}) = p(\mathbf{y}_{0:k-1} \,|\, \boldsymbol{\theta}) \int_{\mathbb{R}^n} p(\mathbf{y}_k \,|\, \mathbf{x}_k, \theta) p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}, \theta) \, d\mathbf{x}_k, \tag{4.2}$$

where the distributions $p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta})$ are given by the Bayesian filtering equations. Like for the Bayesian filtering equations, the recursion for the parameter likelihood is generally intractable and some approximations must be used.

Instead of the full parameter posterior distribution, summary statistics from the posterior distribution are often reported in practice. Typical summary statistics are the mean

$$\bar{\boldsymbol{\theta}} = \mathbb{E}[\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}] = \int_{\mathbb{R}^d} \boldsymbol{\theta} p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}) \, d\boldsymbol{\theta}, \tag{4.3}$$

covariance matrix

$$\mathbb{E}[(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^T \,|\, \mathbf{y}_{0:t}] = \int_{\mathbb{R}^d} (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^T \, d\boldsymbol{\theta} \tag{4.4}$$

and maximum a posteriori (MAP) estimate

$$\hat{\boldsymbol{\theta}}^{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}). \tag{4.5}$$

4.2 Particle Markov chain Monte Carlo

In this section, a sampling based method for approximating the posterior $p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t})$ is presented. In the case of non-linear state-space models, the parameter posterior distribution is generally not available for direct sampling, but can be sampled using Markov chain Monte Carlo (MCMC) based methods. MCMC methods are used to draw samples from a target probability distribution by generating a Markov chain $\{\boldsymbol{\theta}^i\}$ that has the target distribution as a stationary distribution.

The Metropolis-Hastings algorithm is a popular MCMC method, where the i:th sample in the Markov chain is generated by first sampling a candidate $\boldsymbol{\theta}^*$ from a proposal distribution $q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}^{i-1})$. The candidate is then accepted with probability given by

$$\alpha = \min \left\{ 1, \frac{p(\mathbf{y}_{0:t} \,|\, \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*)}{p(\mathbf{y}_{0:t} \,|\, \boldsymbol{\theta}^{i-1}) p(\boldsymbol{\theta}^{i-1})} \frac{q(\boldsymbol{\theta}^{i-1} \,|\, \boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^* \,|\, \boldsymbol{\theta}^{i-1})} \right\}. \tag{4.6}$$

Computing the acceptance probability requires evaluation of the likelihood term $p(\mathbf{y}_{0:t} \,|\, \boldsymbol{\theta})$, which in practice cannot be computed exactly. However, if the likelihood terms are replaced by unbiased estimates, the resulting algorithm still has the same stationary distribution as the exact method [3, 16]. In the case of state-space models, particle filters can be used to compute unbiased estimates for the likelihood, and the resulting method is called particle MCMC (PMCMC) [5]. A particle Metropolis-Hastings algorithm is presented in Algorithm 6. Note that the algorithm is presented only for sampling the parameter $\boldsymbol{\theta}$ from the corresponding marginal posterior distribution. With some modifications, the algorithm can be also used to sample the full joint posterior of $(\mathbf{x}_{0:t}, \boldsymbol{\theta})$, see e.g. [5] for details.

The efficiency of the particle Metropolis-Hastings algorithm depends on the quality of the proposal density $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{i-1})$ and the variance of the likelihood estimates. It should be noted that in general lower variance for the likelihood estimates does not always guarantee better performance, but instead stronger condition of convex stochastic order could be used when comparing PMCMC methods [4].

Constructing an optimal proposal density is generally a difficult task, but sufficient performance can often be obtained by choosing a Gaussian proposal given by

$$q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{i-1}) = \mathrm{N}(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{i-1}, \Sigma). \tag{4.7}$$

Especially for multivariate models, it is often helpful to also let the covariance matrix Σ depend on the chain index i and to use an adaptive method for tuning Σ^i (see e.g. [56]).

Algorithm 6 Particle marginal Metropolis-Hastings

1: Sample $\boldsymbol{\theta}^0 \sim p(\boldsymbol{\theta})$
2: Run a particle filter to get an estimate $\hat{p}(\mathbf{y}_{0:t} \mid \boldsymbol{\theta}^0)$
3: **for** $i \geq 1$ **do**
4: Sample $\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}^{i-1})$
5: Run a particle filter to get an estimate $\hat{p}(\mathbf{y}_{0:t} \mid \boldsymbol{\theta}^*)$
6: Compute acceptance ratio

$$\alpha \leftarrow \min \left\{ 1, \frac{\hat{p}(\mathbf{y}_{0:t} \mid \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*)}{\hat{p}(\mathbf{y}_{0:t} \mid \boldsymbol{\theta}^{i-1}) p(\boldsymbol{\theta}^{i-1})} \frac{q(\boldsymbol{\theta}^{i-1} \mid \boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{i-1})} \right\}$$

7: Sample $u \sim \mathrm{Uniform}(0, 1)$
8: **if** $u < \alpha$ **then**
9: Set $\boldsymbol{\theta}^i \leftarrow \boldsymbol{\theta}^*$ and $\hat{p}(\mathbf{y}_{0:t} \mid \boldsymbol{\theta}^i) \leftarrow \hat{p}(\mathbf{y}_{0:t} \mid \boldsymbol{\theta}^*)$
10: **else**
11: Set $\boldsymbol{\theta}^i \leftarrow \boldsymbol{\theta}^{i-1}$ and $\hat{p}(\mathbf{y}_{0:t} \mid \boldsymbol{\theta}^i) \leftarrow \hat{p}(\mathbf{y}_{0:t} \mid \boldsymbol{\theta}^{i-1})$
12: **end if**
13: **end for**

4.3 Twisted particle filters

Twisted particle filters are best described by considering the full sampling law $M(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ for particles $\mathbf{x}_k = \{\mathbf{x}_k^1, \ldots, \mathbf{x}_k^N\}$ at time k. For the standard bootstrap

particle filter the sampling law is given by

$$M_0(\mathbf{x}_0) = \prod_{i=1}^{N} \mu_0(\mathbf{x}_0^i) \tag{4.8}$$

$$M_k(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) = \prod_{i=1}^{N} \frac{\sum_{i=1}^{N} g(\mathbf{y}_{k-1} \,|\, \mathbf{x}_{k-1}^i) f(\mathbf{x}_k^i \,|\, \mathbf{x}_{k-1}^i)}{\sum_{i=1}^{N} g(\mathbf{y}_{k-1} \,|\, \mathbf{x}_{k-1}^i)}, \tag{4.9}$$

where M_0 and M_k are now probability distributions for the full set of particles. The notation M_0 indicates that particles are sampled from the initial distribution μ_0. Note that sampling from M_k includes also the resampling step.

Twisted particle filters, first presented in [58], are defined by adding a weighting function to the sampling laws of Equations (4.8) and (4.9). This weighted sampling law can be written as

$$\widetilde{M}_0(\mathbf{x}_0) \propto \frac{1}{N} \sum_{s=1}^{N} M_0(\mathbf{x}_0) \psi_0(\mathbf{x}_0^s) \tag{4.10}$$

$$\widetilde{M}_k(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) \propto \frac{1}{N} \sum_{s=1}^{N} M_k(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) \psi_k(\mathbf{x}_{0:k}^s), \tag{4.11}$$

where the functions ψ_k are called twisting functions. In [PIV] the twisted particle filters are presented in a more general form, including also different resampling schemes and more general proposal distributions. For simplicity, only the twisted bootstrap particle filter is considered in this introduction.

The weighted sampling rule can be written in the form

$$\begin{aligned}
\widetilde{M}_0(\mathbf{x}_0) &= \frac{1}{N} \sum_{s=1}^{N} \left[\prod_{i \neq s} \mu_0(\mathbf{x}_0^i) \right] \tilde{\mu}_0(\mathbf{x}_0^s), \\
\widetilde{M}_k(\mathbf{x}_k \,|\, \mathbf{x}_{0:k-1}) &= \frac{1}{N} \sum_{s=1}^{N} \left[\prod_{i \neq s} \frac{\sum_{j=1}^{N} v_{k-1}^j f(\mathbf{x}_k^i \,|\, \mathbf{x}_{k-1}^j)}{\sum_{j=1}^{N} v_{k-1}^j} \right] \\
&\quad \cdot \left[\frac{\sum_{j=1}^{N} \tilde{v}_{k-1}^j \tilde{q}(\mathbf{x}_k^s \,|\, \mathbf{x}_{0:k-1}^j)}{\sum_{j=1}^{N} \tilde{v}_{k-1}^j} \right],
\end{aligned} \tag{4.12}$$

where

$$\tilde{\mu}_0(\mathbf{x}_0) \propto \mu_0(\mathbf{x}_0) \phi_0(\mathbf{x}_0), \tag{4.13}$$

$$\tilde{q}(\mathbf{x}_k \,|\, \mathbf{x}_{0:k-1}^j) \propto f(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}^j) \psi_k(\mathbf{x}_k, \mathbf{x}_{0:k-1}^j), \tag{4.14}$$

and

$$\tilde{v}_{k-1}^{j} = g(\mathbf{y}_{k-1} \mid \mathbf{x}_k^j) \int_{\mathbb{R}^n} f(\mathbf{x}_k \mid \mathbf{x}_{k-1}^j) \psi_k(\mathbf{x}_k, \mathbf{x}_{0:k-1}^j)\, d\mathbf{x}_k. \tag{4.15}$$

From the presentation in Eq. (4.12) we can see that, in the twisted sampling law, the twisting is actually applied to only one of the particles and all the other particles are propagated according to the bootstrap sampling law. The twisted particle filter is summarized in Algorithm 7.

The most important properties of the twisted particle filters are given in the following list. Proofs of these two properties are given in [PIV] for the more general particle filters.

1. Unbiased estimates $\tilde{Z}_k$ for the likelihood $p(\mathbf{y}_{0:k})$ can be computed by recursively setting

$$\tilde{Z}_0 = \frac{\int_{\mathbb{R}^n} \mu_0(\mathbf{x}_0)\psi_0(\mathbf{x}_0)\, d\mathbf{x}_0}{\frac{1}{N}\sum_{j=1}^{N} \psi_0(\mathbf{x}_0^j)} \tag{4.16}$$

$$\tilde{Z}_k = \tilde{Z}_{k-1} \frac{\sum_{j=1}^{N} \tilde{v}_{k-1}^{j}}{\sum_{j=1}^{N} \psi_k(\mathbf{x}_{0:k}^j)}, \quad k > 0. \tag{4.17}$$

   This unbiasedness property is important when twisted particle filters are used as part of the PMCMC algorithm, where it is needed to guarantee that the stationary distribution of the PMCMC sampler is the true posterior distribution of the parameters.

2. The optimal choice for the twisting function (i.e. the choice that minimizes the variance of likelihood estimate $\tilde{Z}_k$) is given by

$$\psi_k(\mathbf{x}_{0:k}) = p(\mathbf{y}_{k:t} \mid \mathbf{x}_k). \tag{4.18}$$

   By choosing twisting function as (4.18), the variance of the likelihood estimate is zero and we get exactly the true values.

In practice the optimal twisting function cannot be readily computed, but the result motivates finding twisting functions that approximate $p(\mathbf{y}_{k:k+l} \mid \mathbf{x}_k)$, where $l \geq 0$ is an additional parameter that controls how many future measurements are used for the approximation. The additional parameter is used for computational reasons since it often happens that beyond a certain value, raising the value of l does not contribute significantly to the accuracy of the estimate of $p(\mathbf{y}_{k:t} \mid \mathbf{x}_k)$ [58], [PIV].

Algorithm 7 Twisted particle filter

1: Sample $K_0 \sim \text{Uniform}(1, \ldots, N)$
2: Sample $\{\mathbf{x}_0^j\}_{j \neq K_0} \sim \mu_0(\mathbf{x}_0^j)$ and $\mathbf{x}_0^{K_0} \sim \tilde{\mu}_0(\mathbf{x}_0^j)$
3: Compute $\tilde{Z}_0$ using Eq. (4.16)
4: **for** $1 \leq k \leq t+1$ **do**
5: Sample $K_k \sim \text{Uniform}(1, \ldots, N)$ and A_k from a discrete distribution with probabilities $\tilde{p}_j \propto \tilde{v}_{k-1}^j$, $1 \leq j \leq N$
6: Sample $\{\mathbf{x}_k^j\}_{j \neq K_k}$ from $f(\mathbf{x}_k^j \,|\, \mathbf{x}_{k-1}^j)$ and sample $\mathbf{x}_k^{K_k} \sim \tilde{q}(\mathbf{x}_k^{K_k} \,|\, \mathbf{x}_{0:k-1}^{A_k})$
7: Compute $\tilde{Z}_k$ using Eq. (4.17)
8: **end for**

4.3.1 Twisted particle filters for Gaussian models

In [PIV] we present methods for implementing the twisted particle filter for non-linear Gaussian models. The presented method is based on approximating the optimal twisting function with

$$p(\mathbf{y}_{k:k+l} \,|\, \mathbf{x}_k) \approx \alpha_{k,l} e^{\mathbf{x}_k^T \Gamma_{k,l} \mathbf{x}_k + \mathbf{x}_k^T \beta_{k,l}}, \tag{4.19}$$

where $\alpha_{k,l} \in \mathbb{R}$, $\beta_{k,l} \in \mathbb{R}^n$ and $\Gamma_{k,l} \in \mathbb{R}^{n \times n}$ are parameters that can depend on the particle history $\mathbf{x}_{0:k-1}$ and the measurements $\mathbf{y}_{k:k+l}$.

In this thesis, two methods for computing the twisting function parameters are presented. The first method is based on local linearization of the nonlinear process and measurement functions around the current state $\mathbf{x}_k^i$ for each of the particles separately. The parameters $\alpha_{k,l}^i$, $\beta_{k,l}^i$ and $\Gamma_{k,l}^i$ for particle i can then be computed by running the EKF algorithm starting from initial state $\mathbf{m}_{k-1} = \mathbf{x}_{k-1}^i$ and $\mathrm{P}_{k-1} = 0$ (see [PIV] and Algorithms 5 and 6).

The second, computationally less intensive method, is obtained by linearising around the mode of $p(\mathbf{y}_{k:k+l} \,|\, \mathbf{x}_k)$ and using the same twisting function parameters for all particles (see [PIV] and Algorithm 7). Finding the mode of $p(\mathbf{y}_{k:k+l} \,|\, \mathbf{x}_k)$ is generally not trivial, but for an unimodal case a reasonable approximation can be obtained using an extended Kalman smoother that is initialised with mean and variance computed using the particle states $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$.

Details for the equations and resulting twisted particle filter can be found from [PIV] (see Algorithm 4).

4.4 Example: estimation of parameters for target tracking model

This numerical example extends the target tracking problem of previous sections by adding the noise variance σ^2 as an unknown parameter. The unknown parameter is estimated using PMCMC method with a standard bootstrap particle filter and a twisted particle filter. The prior distribution for σ^2 is chosen to be an inverse Gamma distribution with shape and scale parameters set to 0.1.

In this example, the twisting function is approximated using linearisation around the mode of $p(\mathbf{y}_{k:k+l} \mid \mathbf{x}_k)$. Approximation for the mode is computed by running a fixed-point extended Kalman smoother (see [49, p. 160]) for the state at time k. The Kalman smoother is initialised with

$$\mathbf{m}_{k-1} = \sum_{i=1}^{N} w_{k-1}^i \mathbf{x}_{k-1}^i,$$
$$\mathrm{P}_{k-1} = \sum_{i=1}^{N} w_{k-1}^i (\mathbf{x}_{k-1}^i - \mathbf{m}_{k-1})(\mathbf{x}_{k-1}^i - \mathbf{m}_{k-1})^T.$$

The choice of a value for the parameter l that determines the number of measurements used at each time step of the twisted particle filter can be studied by plotting the variance of likelihood estimates $\tilde{Z}_t$ for some fixed values of the unknown parameters. For numerical reasons it is often better to deal with logarithms of the likelihood estimates instead of actual values.

Figure 4.1 (a) shows how the variance of $\log \tilde{Z}_t$ behaves with different values of l. It can be seen that the variance reduces fast around $l = 50$, but after that point the additional measurements have less impact. From Figure 4.1 (b) it can be seen that the twisted particle filter produces significantly lower variances compared to the bootstrap particle filter with the same number of particles. As noted earlier, the low variance of the likelihood estimates is crucial for an efficient PMCMC algorithm.

Figure 4.2 shows MCMC chains computed using the PMCMC algorithm for a bootstrap particle filter and a twisted particle filter with the same number of particles. Clearly $N = 100$ particles is too low for the bootstrap particle filter and produces highly correlated samples. The quality of the MCMC chain can be assessed using e.g. the sample autocorrelation function, given by

$$R(\tau) = \frac{1}{R(0)} \frac{1}{M-1} \sum_{j=1}^{M-\tau} (\boldsymbol{\theta}_j - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_{j+\tau} - \bar{\boldsymbol{\theta}}), \tag{4.20}$$

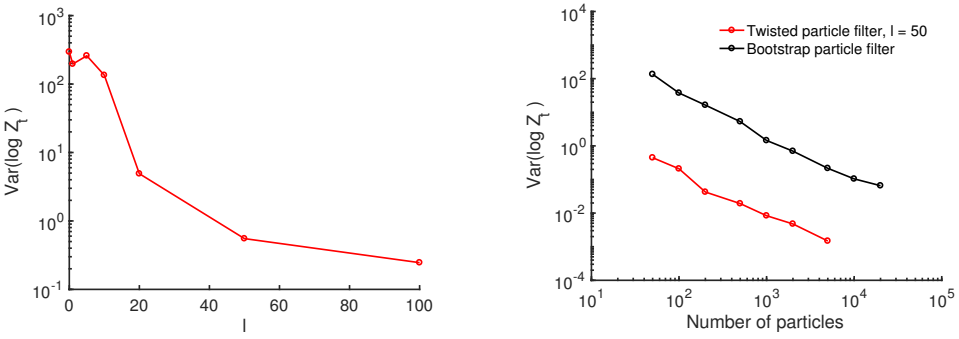


Figure 4.1: Variance of $\log Z_t$ computed from 100 simulations as a function of (a) parameter l and (b) number of particles N.

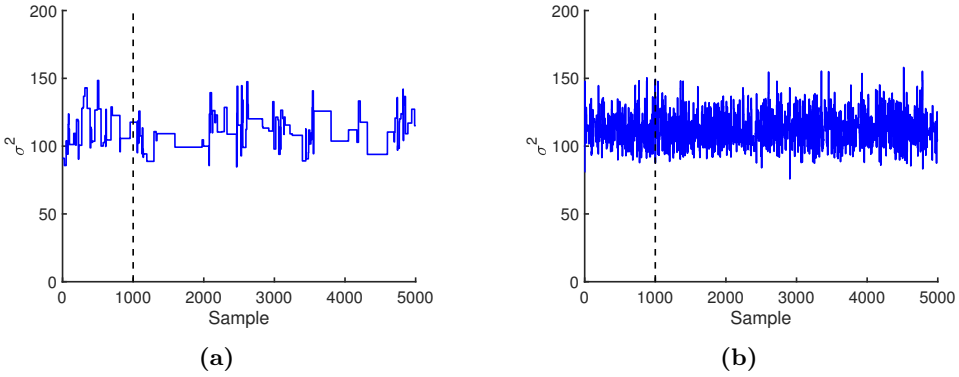


Figure 4.2: Samples from PMCMC algorithm using (a) bootstrap particle filter with $N = 100$ and (b) twisted particle filter with $N = 100$ and $l = 50$. The dashed line shows the chosen burn in point.

where $\bar{\boldsymbol{\theta}}$ is the sample mean and M is the number of samples in the chain. Figure 4.3 shows the sample autocorrelation for the resulting chains. For a good quality chain, the autocorrelation function should quickly approach zero as the lag increases. From the autocorrelation plot it can be seen that the bootstrap particle filter needs over $N = 1500$ particles to obtain similar quality as the twisted particle filter with $N = 100$ and $l = 50$.

Figure 4.4 shows the histograms computed from the PMCMC chains. It can be seen that with $N = 100$ particles, the twisted particle filter produces significantly better estimate for the parameter posterior compared to the bootstrap particle filter.

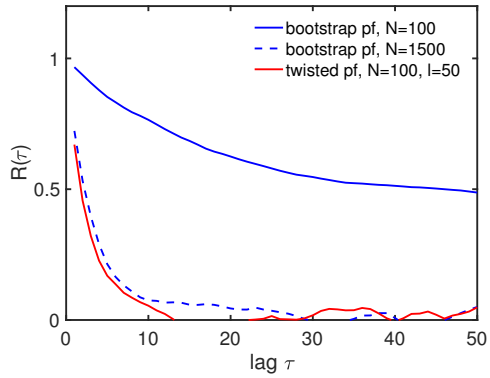


Figure 4.3: Autocorrelation for the PMCMC chain with 5000 samples for the bootstrap particle filter and twisted particle filter.

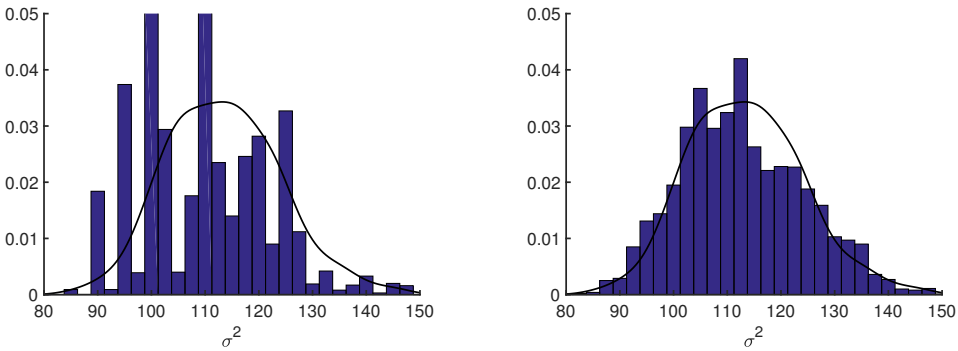


Figure 4.4: Histograms for the (a) bootstrap particle filter with $N = 100$ and (b) twisted particle filter with $N = 100$ and $l = 50$. The histograms are computed from PMCMC chains with 5000 samples, where the first 1000 samples were excluded as burn in period. The black line shows the true posterior pdf estimated from a chain of 20 000 samples computed using bootstrap particle filter PMCMC with 10 000 particles.

5 Conclusion and discussion

This chapter gives conclusions and some discussion for each of the topics presented in the thesis.

Using Gaussian filtering methodology for improved statistical GNSS orbit prediction

The first contribution deals with practical application of the Gaussian filtering methodology to a GNSS satellite orbit prediction problem. The problem is solved using extended Kalman filter, and sigma-point based unscented Kalman filter and cubature Kalman filter.

For this problem, there was no major difference in performance between the extended Kalman filter and sigma-point based filters. This indicates that the nonlinearities are not severe enough to benefit from the more sophisticated approximations. Also, an algorithm for estimating the process noise variance was presented. It was found that using a suitable variance for the process noise, that is estimated from the data, the consistency of the predicted variance of the satellite's orbit can be significantly improved. The results of the paper have been used in the development of a commercial product, where the satellite's predicted orbit is used to reduce the time to first fix in a GNSS positioning device (see also [53]).

Developing online algorithms for statistical inference in regression and filtering problems with robust measurement noise models

In practical applications the measurement noise is often assumed to be Gaussian, since this allows the use of well known methods for fast approximate inference. If the noise in reality differs significantly from the Gaussian assumption, this leads to bad approximation of the posterior distribution and erroneous inference results. A typical case is the presence of outliers in the measurement data. Gaussian distribution has a very low probability for observations far away from the mean, which results

in outliers having a too large impact to the resulting posterior distribution. The outliers can be taken into account by using a measurement distribution having more probability in the tails than the Gaussian distribution. The heavier the tails, the less effect the outliers have on the inference.

In this thesis, the robust inference problem is solved using a factorized form of the variational Bayes algorithm, where the intractable posterior is approximated to factorize to a product of independent components. The functional form and parameters of the components are determined by minimizing the Kullback-Leibler divergence of the approximating distribution with respect to the true posterior distribution. This approach can be used for robust distributions that can be presented as Gaussian scale mixtures, where an extra latent variable is used to scale the variance of the Gaussian distribution. An algorithm for robust linear regression is derived for a large family of Gaussian scale mixture models having e.g. the t-distribution, Laplace distribution and contaminated normal distribution as special cases. An open-source Matlab implementation has been published. The introductory part of the thesis also presents how the regression algorithm can be extended to be used in Kalman-like robust filtering algorithm.

Improving Gaussian filtering based smoothing for continuous-discrete Gaussian models using iterative variational Gaussian algorithm

Variational Bayes methodology can be also used to form Gaussian approximations. In this case, the functional form of the approximating distribution is fixed to be Gaussian, where the mean and variance are chosen to minimize the Kullback-Leibler divergence as in the factorised case. This approach can be used to approximate the smoothing distribution of continuous-discrete Gaussian models, where the process is modeled by continuous time stochastic differential equations and measurements are discrete in time. In the thesis, the existing VB smoothing results presented by Archambeau *et al.* [10 – 12] are extended to more general models. A comparison and connection to the Gaussian filtering based smoothing is also presented.

The variational Bayes method is a versatile tool for approximating intractable posterior distributions, but can be challenging to apply in practice since the equations are often complicated and prone to implementation errors. A user friendly library or toolbox would greatly facilitate the use of VB based methods in practice.

Another problem with the VB based methods, and with most deterministic approximation methods, is that they do not provide any guarantees for the quality of the

approximation. In contrast, sampling based particle filters have well established theoretical results for the convergence, but the filters can be computationally too heavy for real time applications with low end devices. An interesting research topic is using the deterministic approximations as a part of sampling based methods to make them more efficient and reduce the computational cost. For example, the VB results for the robust measurement model could be used to form the proposal distribution for the particles in an outlier robust particle filter algorithm.

Improving efficiency of particle Markov chain Monte Carlo methods using twisted particle filters and Gaussian filtering methodology

Finally, this thesis presents results for estimating static parameters in the state-space model using sampling based MCMC method. Particle MCMC methods are used to generate samples from the parameter posterior distribution by using a particle filter as part of the MCMC algorithm. A popular choice is the Metropolis-Hastings algorithm, where the particle filter is used to compute unbiased estimates for the likelihood term in the formula for acceptance probability. The PMCMC is computationally extremely heavy, since a particle filter must be run at each iteration and typically thousands of iterations are needed for sufficient approximation of the parameter posterior distribution. Twisted particle filters were introduced by Whiteley *et al.* [58] to specifically provide an optimal estimate of the likelihood term. In this thesis, it is shown how the twisted particle filter can be used in the PMCMC algorithm to get computation gains against standard particle filters.

The practical implementation of the twisted particle filter requires approximating the so called optimal twisting function, which is used to weight the sampling law for the particles and is the basis for the optimality of the likelihood estimate. This thesis considers forming the approximation for non-linear Gaussian state-space models, where Gaussian filtering methodology can be used to compute the approximation. The most significant computational advantages were obtained when the optimal twisting function could be assumed to be unimodal, which enabled use of a global approximation for the twisting function. This might not be the case in many practical applications and more research would be needed for formulating a more generally applicable twisted particle filtering algorithm. One possibility could be using parallel computation and the local linearization approach, where an extended Kalman filter is run for each particle separately. The local linearization approach doesn't rely on the unimodal assumption and was found to provide significant increase in efficiency in terms of the number of particles required for a specific

likelihood estimate variance. However, sequentially computing the required extended Kalman filtering steps for each particle is computationally very heavy.

Bibliography

[1] Agamennoni, G., Nieto, J. I., and Nebot, E. M., "An outlier-robust Kalman filter," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1551–1558.

[2] Andrews, D. F. and Mallows, C. L., "Scale mixtures of normal distributions," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 99–102, 1974.

[3] Andrieu, C. and Roberts, G. O., "The pseudo-marginal approach for efficient Monte Carlo computations," *The Annals of Statistics*, vol. 37, no. 2, pp. 697–725, 2009.

[4] Andrieu, C. and Vihola, M., "Establishing some order amongst exact approximations of mcmcs," *Ann. Appl. Probab.*, vol. 26, no. 5, pp. 2661–2696, 10 2016. [Online]. Available: https://doi.org/10.1214/15-AAP1158

[5] Andrieu, C., Doucet, A., and Holenstein, R., "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[6] Arasaratnam, I. and Haykin, S., "Cubature Kalman filters," *Automatic Control, IEEE Transactions on*, vol. 54, no. 6, pp. 1254–1269, 2009.

[7] ———, "Cubature Kalman smoothers," *Automatica*, vol. 47, no. 10, pp. 2245–2250, 2011.

[8] Arasaratnam, I., Haykin, S., and Elliott, R. J., "Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[9] Arasaratnam, I., Haykin, S., and Hurd, T. R., "Cubature Kalman filtering for continuous-discrete systems: theory and simulations," *Signal Processing, IEEE Transactions on*, vol. 58, no. 10, pp. 4977–4993, 2010.

[10] Archambeau, C. and Opper, M., "Approximate inference for continuous-time Markov processes," in *Bayesian Time Series Models*. Cambridge University Press, 2011, pp. 125–140.

[11] Archambeau, C., Cornford, D., Opper, M., and Shawe-Taylor, J., "Gaussian process approximations of stochastic differential equation," *Journal of Machine Learning Research: Workshop and Conference Proceedings*, vol. 11, pp. 1–16, 2007.

[12] Archambeau, C., Opper, M., Shen, Y., Cornford, D., and Shawe-taylor, J. S., "Variational inference for diffusion processes," in *Advances in Neural Information Processing Systems 20*, Platt, J., Koller, D., Singer, Y., and Roweis, S., Eds. Curran Associates, Inc., 2008, pp. 17–24.

[13] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, 2004.

[14] Barber, D., *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[15] Beal, M. J., "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University of London, 2003.

[16] Beaumont, M. A., "Estimation of population growth or decline in genetically monitored populations." *Genetics*, vol. 164, no. 3, pp. 1139–1160, 07 2003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1462617/

[17] Bishop, C. M., *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] Briers, M., Doucet, A., and Maskell, S., "Smoothing algorithms for state–space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010.

[19] Casella, C. P. and Casella, G., *Monte Carlo Statistical methods*. Springer, 1999.

[20] Cox, H., "On the estimation of state variables and parameters for noisy dynamic systems," *Automatic Control, IEEE Transactions on*, vol. 9, no. 1, pp. 5 – 12, 1964.

[21] Doucet, A., Godsill, S., and Andrieu, C., "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197 – 208, 2000.

[22] Doucet, A., De Freitas, N., and Gordon, N., *Sequential Monte Carlo methods in practice*. Springer, 2001.

[23] Fraser, D. and Potter, J., "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387 – 390, Aug 1969.

[24] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B., *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC, 2013.

[25] Gordon, N. J., Salmond, D. J., and Smith, A. F., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 2. IET, 1993, pp. 107 – 113.

[26] Ito, K. and Xiong, K., "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910 – 927, 2000.

[27] Jazwinski, A. H., *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[28] Julier, S. J. and Uhlmann, J. K., "New extension of the Kalman filter to nonlinear systems," in *AeroSense'97*. International Society for Optics and Photonics, 1997, pp. 182 – 193.

[29] – – –, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401 – 422, 2004.

[30] Kalman, R. E. and Bucy, R. S., "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95 – 108, 1961.

[31] Kalman, R. E., "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35 – 45, 1960.

[32] Kalnay, E., *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 2003.

[33] Keeling, M. J. and Rohani, P., *Modeling infectious diseases in humans and animals*. Princeton University Press, 2008.

[34] Kitagawa, G., "Non-gaussian state-space modeling of nonstationary time series," *Journal of the American statistical association*, vol. 82, no. 400, pp. 1032–1041, 1987.

[35] ———, "The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother," *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 4, pp. 605–623, 1994.

[36] Kotz, S. and Nadarajah, S., *Multivariate t distributions and their applications*. Cambridge University Press, 2004.

[37] Kotz, S., Kozubowski, T. J., and Podgórski, K., *The Laplace distribution and generalizations*. Birkhäuser, 2001.

[38] Kullback, S. and Leibler, R. A., "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. pp. 79–86, 1951.

[39] Lange, K. and Sinsheimer, J. S., "Normal/independent distributions and their applications in robust regression," *Journal of Computational and Graphical Statistics*, vol. 2, no. 2, pp. pp. 175–198, 1993. [Online]. Available: http://www.jstor.org/stable/1390698

[40] MacKay, D. J. C., *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[41] Montenbruck, O. and Gill, E., *Satellite Orbits: Models, Methods and Applications*, corrected 3rd printing ed. Springer, 2005.

[42] Murphy, K. P., *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[43] Øksendal, B., *Stochastic differential equations*. Springer, 2003.

[44] Piché, R., Särkka, S., and Hartikainen, J., "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate student-t distribution," in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*. IEEE, 2012, pp. 1–6.

[45] Pitt, M. K. and Shephard, N., "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[46] Rauch, H. E., Tung, F., and Striebel, C. T., "Maximum likelihood estimates of linear dynamic systems," *Journal of American Institute of Aeronautics and Astronautics (AIAA)*, vol. 3, no. 8, pp. 1445–1450, 1965.

[47] Ristic, B., Arulampalm, S., and Gordon, N. J., *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

[48] Särkkä, S., "Unscented Rauch–Tung–Striebel smoother," *Automatic Control, IEEE Transactions on*, vol. 53, no. 3, pp. 845–849, 2008.

[49] ———, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[50] Särkkä, S. and Hartikainen, J., "Sigma point methods in optimal smoothing of non-linear stochastic state space models," in *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2010, pp. 184–189.

[51] Särkkä, S. and Sarmavuori, J., "Gaussian filtering and smoothing for continuous-discrete dynamic systems," *Signal Processing*, vol. 93, no. 2, pp. 500–510, Feb. 2013.

[52] Sarmavuori, J. and Särkkä, S., "Fourier-Hermite Kalman filter," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1511–1515, 2012.

[53] Seppänen, M., Ala-Luhtala, J., Piché, R., Martikainen, S., and Ali-Löytty, S., "Autonomous prediction of GPS and GLONASS satellite orbits," *NAVIGATION*, vol. 59, no. 2, pp. 119–134, 2012.

[54] Tukey, J. W., "A survey of sampling from contaminated distributions," in *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. Stanford University Press, 1960, pp. 448–485.

[55] Van Der Merwe, R., Doucet, A., De Freitas, N., and Wan, E., "The unscented particle filter," in *NIPS*, 2000, pp. 584–590.

[56] Vihola, M., "Robust adaptive Metropolis algorithm with coerced acceptance rate," *Statistics and Computing*, vol. 22, no. 5, pp. 997–1008, 2012.

[57] Wan, E., Van Der Merwe, R. *et al.*, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000.* IEEE, 2000, pp. 153–158.

[58] Whiteley, N. and Lee, A., "Twisted particle filters," *The Annals of Statistics*, vol. 42, no. 1, pp. 115–141, 2014.

[59] Wilkinson, D., *Stochastic Modelling for System Biology.* CRC Press, 2006.

[60] Wu, Y., Hu, D., Wu, M., and Hu, X., "A numerical-integration perspective on Gaussian filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2910–2921, 2006.

Publications

Publication 1

Juha Ala-Luhtala, Mari Seppänen, Simo Ali-Löytty, Robert Piché and Henri Nurminen, Estimation of initial state and model parameters for autonomous GNSS orbit prediction, *In Proceedings of International Global Navigation Satellite Systems Society Symposium 2013 (IGNSS2013)*, 2013

International Global Navigation Satellite Systems Society
IGNSS Symposium 2013

Outrigger Gold Coast, Qld Australia
16-18 July, 2013

Estimation of initial state and model parameters for autonomous GNSS orbit prediction

Juha Ala-Luhtala
Tampere University of Technology/Finland
Phone: +358456509464, juha.ala-luhtala@tut.fi
Mari Seppänen
Tampere University of Technology/Finland
mari.j.seppanen@tut.fi
Simo Ali-Löytty
Tampere University of Technology/Finland
simo.ali-loytty@tut.fi
Robert Piché
Tampere University of Technology/Finland
robert.piche@tut.fi
Henri Nurminen
Tampere University of Technology/Finland
henri.nurminen@tut.fi

ABSTRACT

In self-assisted GNSS the orbit of a satellite is predicted by solving the differential equation that models its motion. Our motion model includes the most important forces: Earth's gravity, lunar and solar gravity and solar radiation pressure. Unmodeled forces are taken into account by using Gaussian white noise term with covariance matrix estimated offline from historical orbital data. The estimation of model parameters (solar radiation pressure and Earth orientation parameters) and initial state for the prediction includes both offline and online stages. In the offline stage, priors for the solar radiation pressure parameters are estimated using precise orbits issued by the International GNSS service (IGS). In the online stage, the satellite's broadcast ephemeris is used to estimate the initial state and model parameters. The estimation of the initial state is formulated as non-linear continuous-time filtering problem with discrete-time measurements. The filtering equations are solved numerically and the performance of different numerical methods (Extended, Cubature and Unscented Kalman filters) is compared. Using the estimated initial state and model parameters, the satellite orbits are predicted 5 days into the future. The accuracy and consistency of the predicted orbits is analysed by comparing with the IGS precise ephemerides. In this paper only GPS satellites are considered, but the method can be extended to other satellite systems.

KEYWORDS: Satellite orbit prediction; Gaussian filtering; Estimation

1. INTRODUCTION

In autonomous or "self-assisted" GNSS orbit prediction the orbit is predicted in the positioning device using only information from the satellite's broadcast ephemeris. The predicted orbit can be used for example to reduce the time to first fix (TTFF) of a standalone GNSS receiver (Mattos, 2008), (Zhang *et al*., 2008), (Lytvyn *et al*., 2012).

Like most GNSS prediction algorithms, our algorithm is based on integrating the satellite's equation of motion several days forward using initial conditions computed from the satellite's broadcast ephemeris. Our equation of motion includes the four most significant forces acting on the satellite: the gravitations of the Earth, the Sun and the Moon, and the solar radiation pressure. The models for the gravitational terms are covered in (Seppänen *et al*. 2011), (Seppänen *et al*., 2012). Our two-parameter solar radiation pressure model is presented in (Ala-Luhtala *et al*., 2012). The uncertainty caused by modelling errors and unmodeled forces are taken into account by adding a Gaussian white noise acceleration term to the satellite's equation of motion. The covariance matrix for the Gaussian white noise term is estimated using historical precise ephemeris data.

One of the main problems in the autonomous orbit prediction is obtaining the initial conditions needed to start the integration. The broadcast position and velocity are given in the Earth-centered, Earth-fixed reference frame and must be transformed to an inertial reference frame. However, the Earth orientation parameters (EOP) that are needed in the transformation are unknown. We have also noticed that the velocity computed directly from the broadcast ephemeris is not accurate enough for prediction. In our previous work, we solved the initial condition determination problem by fitting the motion model to the broadcast data using an iterative least-squares minimization algorithm (Seppänen *et al.*, 2011), (Seppänen *et al*., 2012).

In this paper we propose a Bayesian filtering algorithm for the determination of the initial state. The filtering solution has the advantage that it requires only one iteration and enables a probability-based interpretation of the problem. The filtering problem we are considering in this paper is nonlinear with continuous-time process model and discrete-time measurements. The exact solution is analytically intractable and numerical approximations are used. We consider the approximate solution obtained using the Extended Kalman Filter (Jazwinski, 1970), which is based on linearization using a first-order Taylor polynomial. We also consider the Unscented Kalman Filter (UKF) (Julier *et al*., 1995) and the Cubature Kalman Filter (CKF) (Arasaratnam and Haykin, 2009), which are based on sigma-point approximations for the statistical moments needed in the filtering algorithm.

The paper is organized as follows. The satellite's equation of motion and the reference frames are covered in Sections 2.1 and 2.2. In Section 2.3 a method for estimating the process noise covariance matrix using precise ephemeris data is presented. The method for estimating the initial state using the broadcast ephemeris is presented in Section 2.4. Section 2.5 summarizes the proposed orbit prediction algorithm. The algorithms prediction accuracy is assessed in Chapter 3. The paper closes with conclusion and discussion in Chapter 4.

2. MODEL

2.1 Equation of motion

The four most significant forces affecting the satellite are the gravitational attractions of the Earth, the Sun and the Moon, and the solar radiation pressure (srp). The acceleration of the satellite is

$$\boldsymbol{a}_{\mathrm{sat}} = \boldsymbol{a}_{\mathrm{earth}} + \boldsymbol{a}_{\mathrm{sun}} + \boldsymbol{a}_{\mathrm{moon}} + \boldsymbol{a}_{\mathrm{srp}}, \tag{1}$$

where all the accelerations are in an inertial reference frame.

The acceleration caused by the Earth is computed by taking the gradient of the gravity potential U. To account for the uneven mass distribution of the Earth, the gravity potential is written using a spherical harmonic series (Montenbruck and Gill, 2005). Seppänen (2010) found that at GNSS satellite altitudes, terms up to the order of at least 4 should be used. In our implementation, we have used terms up to the degree and order 8.

The gravitational acceleration caused by any celestial body can be computed using equation

$$\mathbf{a}_{\mathrm{cb}} = GM\left(\frac{\mathbf{r}_{\mathrm{cb}} - \mathbf{r}}{\|\mathbf{r}_{\mathrm{cb}} - \mathbf{r}\|^{\mathbf{3}}} - \frac{\mathbf{r}_{\mathrm{cb}}}{\|\mathbf{r}_{\mathrm{cb}}\|^{\mathbf{3}}}\right), \tag{2}$$

where M is the mass of the body, $\mathbf{r_{cb}}$ is its position in Earth centered inertial reference frame, and $\mathbf{r}$ is the position of the satellite in the same reference frame. The orbits of the sun and the moon are computed using simple models presented by Montenbruck and Gill (2005. See references (Seppänen et al, 2011, Seppänen et al, 2012) for more details about the computation of the gravity terms.

For the acceleration caused by the solar radiation pressure, we use a two-parameter empirical model (Ala-Luhtala *et al*., 2012)

$$\mathbf{a}_{\mathrm{srp}} = \lambda\left(-\frac{\alpha_1 C}{r_{\mathrm{sun}}^2}\mathbf{e}_s + \alpha_2 \mathbf{e}_y\right). \tag{3}$$

The first term inside the parenthesis describes the effect of direct solar radiation pressure, which is directed along the line joining the satellite and the sun. The magnitude of the direct solar radiation pressure depends on the satellite's distance to the sun r_{sun}. The term α_1 is a satellite specific parameter that scales the direct radiation pressure, and C is a known constant common for all the satellites. The second term inside the parenthesis models the so called y-bias acceleration, which is directed along the satellites solar panel axis (Springer *et al*., 1999), (Froideval, 2009). The y-bias parameter α_2 is also satellite specific. To account for the shadowing of the Earth, we use a time varying term λ that is based on the conical shadow model described by Montenbruck and Gill (2005). The times when the satellite enters Earth's shadow are called eclipse seasons.

In addition to the previously described four forces, there are numerous smaller forces acting on the satellite. These include for example the gravitation of other celestial bodies, the

radiation pressure of the proportion of the incident sunlight that is reflected by the Earth (albedo), and Earth tides (Montenbruck and Gill, 2005). We do not attempt to model the forces accurately, but instead take them into account by adding a stochastic acceleration term to Eq. (1). In this paper, we use a Gaussian white noise model. Formally, we can write the new acceleration equation as

$$\mathbf{a} = \mathbf{a}_{\text{sat}} + \mathbf{L}(\mathbf{r}, \mathbf{v})\mathbf{w}, \tag{4}$$

where $\mathbf{a}_{\text{sat}}$ is computed using Eq. (1) and $\mathbf{w}$ is a Gaussian white noise stochastic process with zero mean and covariance matrix $\mathbf{Q_a}$. We have chosen to use the satellite centered RTN-coordinate system (Radial, Tangential, Normal) for the white noise term. The transformation matrix **L(r, v)** transforms the white noise into the inertial reference frame. The transformation from Earth centered inertial (ECI) reference system to the RTN system is given by (Tapley *et al*., 2004)

$$\mathbf{r}_{\text{RTN}} = [\mathbf{e}_R \quad \mathbf{e}_T \quad \mathbf{e}_N]\mathbf{r}_{\text{ECI}}, \tag{5}$$

where the unit vectors are

$$\mathbf{e}_R = \frac{\mathbf{r}_{\text{ECI}}}{\|\mathbf{r}_{\text{ECI}}\|}, \tag{6}$$

$$\mathbf{e}_T = \mathbf{e}_N \times \mathbf{e}_R, \tag{7}$$

$$\mathbf{e}_N = \frac{\mathbf{r}_{\text{ECI}} \times \mathbf{v}_{\text{ECI}}}{\|\mathbf{r}_{\text{ECI}} \times \mathbf{v}_{\text{ECI}}\|}. \tag{8}$$

Eq. (4) should be interpreted as a first order Itô stochastic differential equation

$$d\mathbf{x} = \mathbf{f}(\mathbf{x})dt + \mathbf{D}(\mathbf{x})d\boldsymbol{\beta}_{\mathbf{a}}, \tag{9}$$

where

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{r}, \mathbf{v}) = \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{r}) \end{bmatrix}, \tag{10}$$

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{r}, \mathbf{v}) = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{L}(\mathbf{r}, \mathbf{v}) \end{bmatrix} \tag{11}$$

and $\boldsymbol{\beta}_{\mathbf{a}}$ is a 3-dimensional Brownian motion with diffusion matrix $\mathbf{Q_a}$. The state $\mathbf{x}$ includes the position and velocity of the satellite in the inertial reference frame. See for example references (Øksendal, 2003) and (Jazwinski, 1970) for more information about stochastic differential equations.

2.2 Reference frames

An Earth-fixed, Earth-centered (ECEF) system has its origin at the mass center of the Earth and its axes are fixed with respect to the Earth's surface. In GPS, the reference frame is WGS84, which, for our purposes, can be considered equal to the Terrestrial Reference System (TRS) maintained by the International Earth Rotation and Reference Systems Service (IERS). The origin of the TRS system is the Earth's centre of mass and its z-axis is the mean rotational

axis of the Earth.

An inertial reference system maintained by the IERS is the Celestial Reference System (CRS). CRS is a reference system whose coordinate axes maintain their orientation with respect to distant stars. The origin of this reference frame is the center of the Earth and Earth is in an accelerated motion while orbiting around the sun. Therefore CRS is not precisely inertial, but is an adequate approximation of an inertial reference frame for our purposes. The transformation from the TRS system at epoch *t* to the CRS system is

$$\mathbf{r}_{\mathrm{TRS}}(t) = \mathbf{W}(t)\mathbf{G}(t)\mathbf{N}(t)\mathbf{P}(t)\mathbf{r}_{\mathrm{CRS}}, \tag{12}$$

where the matrices **W**, **G**, **N** and **P** describe polar motion, Earth rotation, nutation, and precession, respectively. See references (Seppänen *et al*., 2011) and (Seppänen *et al*, 2012) for details on the computation of matrices **G**, **N** and **P**. The polar motion matrix **W** is described using equation

$$\mathbf{W}(t) = \mathbf{R}_y\big(-x_p(t)\big)\mathbf{R}_x\big(-y_p(t)\big), \tag{13}$$

where x_p and y_p are the polar motion parameters and $\mathbf{R}_x$ and $\mathbf{R}_y$ are rotation matrices about the x- and y-axes. Together with dUT1, x_p and y_p are also called Earth orientation parameters (EOP). dUT1 is the difference between Universal Time (UT1) and Coordinated Universal Time (UTC). This difference is small and in our implementation we use $\mathrm{dUT1} = 0$. This approximation leads to a median error of 4.2 m in the satellite's position for a 4 day long prediction (Seppänen *et al*., 2012). The daily values for these parameters can be found from the homepage of IERS (IERS, 2013).

Instead of the CRS frame, we choose the inertial reference frame to be an intermediate reference system, denoted by $\mathrm{TIRS}(t_0)$, at time t_0. The transformation from $\mathrm{TIRS}(t_0)$ to TRS is given by

$$\mathbf{r}_{\mathrm{TRS}} = \mathbf{W}(t)\mathbf{G}(t)\mathbf{N}(t)\mathbf{P}(t)\mathbf{P}^T(t_0)\mathbf{N}^T(t_0)\mathbf{G}^T(t_0)\mathbf{r}_{\mathrm{TIRS}(t_0)}. \tag{14}$$

For a prediction of a few days, the nutation and precession matrices remain almost unchanged. That is, we can make the approximations $\mathbf{P}(t)\mathbf{P}^{\mathbf{T}}(t_0) \approx \mathbf{I}$ and $\mathbf{N}(t)\mathbf{N}^{\mathbf{T}}(t_0) \approx \mathbf{I}$. Using these approximations Eq. (14) is reduced to

$$\mathbf{r}_{\mathrm{TRS}} = \mathbf{W}(t)\mathbf{R}_z\big((t-t_0)\omega\big)\mathbf{r}_{\mathrm{TIRS}(t_0)}. \tag{15}$$

We used also the result $\mathbf{G}(t)\mathbf{G}^T(t_0) = \mathbf{R}_z\big((t-t_0)\omega\big)$, where ω is the angular velocity of the Earth's rotation (Seppänen *et al*., 2012). We use the notation $\mathbf{T}(t) = \mathbf{W}(t)\mathbf{R}_z\big((t-t_0)\omega\big)$ for the transformation matrix from $\mathrm{TIRS}(t_0)$ to TRS.

The transformation for the velocity can be derived by differentiating Eq. (15) with respect to time. This gives

$$\mathbf{v}_{\mathrm{TRS}} = \mathbf{T}(t)\mathbf{v}_{\mathrm{TIRS}(t_0)} - \boldsymbol{\omega}\times\big(\mathbf{T}(t)\mathbf{r}_{\mathrm{TIRS}(t_0)}\big), \tag{16}$$

where $\boldsymbol{\omega} = [0 \quad 0 \quad \omega]^T$ is the angular velocity vector of the Earth's rotation.

2.3 Offline estimation of the process noise covariance

For the process noise covariance we use a diagonal matrix

$$\mathbf{Q_a} = \begin{bmatrix} e^{q_R} & 0 & 0 \\ 0 & e^{q_T} & 0 \\ 0 & 0 & e^{q_N} \end{bmatrix}, \tag{17}$$

where the diagonal elements are the variances in the radial, tangential and normal directions. The exponential parametrization is used for scaling to avoid numerical errors caused by very small values, and also to convert the problem of estimating the parameter values to an unconstrained optimization problem. The variances are estimated using precise ephemeris data from the National Geospatial-intelligence Agency (NGA, 2013). The reason for using NGA instead of IGS precise ephemerides is that from NGA we get also the velocity of the satellite.

Consider that we have the precise ephemerides $\mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_N}$ (i.e. the position and velocity) at times $t_0, t_1, \ldots, t_N$ in an inertial reference frame. The transformation to inertial reference frame can be done using the daily EOP values provided by IERS (IERS, 2013). The posterior distribution for the process noise covariance parameters $\mathbf{q} = [q_R \quad q_T \quad q_N]$ is

$$p(\mathbf{q} \,|\mathbf{x}_0, \ldots, \mathbf{x}_N) \propto p(\mathbf{x}_0, \ldots, \mathbf{x}_N|\, \mathbf{q}) p(\mathbf{q}). \tag{18}$$

The likelihood can be written by

$$p(\mathbf{x}_0, \ldots, \mathbf{x}_N \mid \mathbf{q}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \mathbf{x}_{i-1}, \mathbf{q}) p(\mathbf{x}_0) \tag{19}$$

The conditional probability densities in Eq. (19) could in principle be obtained by discretizing the stochastic differential equation (9) using time step $\Delta t = t_i - t_{i-1}$. However, due to the nonlinear model and state dependent noise term, the conditional densities are difficult to compute. For this reason, we approximate the conditional densities in Eq. (19) with a multivariate normal density

$$p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{q}) = \mathrm{Normal}(\mathbf{x}_i|\mathbf{m}_i, \mathbf{\Sigma}_i(\mathbf{q})) \tag{20}$$

where $\mathbf{m}_i$ and $\mathbf{\Sigma}_i(\mathbf{q})$ are solutions to the differential equations

$$\frac{d\mathbf{m}_i(t)}{dt} = \mathbf{f}(\mathbf{m}(t)) \tag{21}$$

$$\begin{aligned} \frac{d\mathbf{\Sigma}_i(t, \mathbf{q})}{dt} = {} & \mathbf{\Sigma}(t, \mathbf{q}) \mathbf{F}_{\mathbf{x}}^T(\mathbf{m}(t)) + \mathbf{F}_{\mathbf{x}}(\mathbf{m}(t)) \mathbf{\Sigma}(t, \mathbf{q}) \\ & + \mathbf{L}(\mathbf{m}(t)) \mathbf{Q_a}(\mathbf{q}) \mathbf{L}^T(\mathbf{m}(t)) \end{aligned} \tag{22}$$

with initial conditions $\mathbf{m}(0) = \mathbf{x}_{i-1}$ and $\mathbf{\Sigma}(0, \mathbf{q}) = \mathbf{0}$. The matrix $\mathbf{F_x}$ is the Jacobian matrix of $\mathbf{f}$. Note that this is the same approximation made in the prediction step of the Extended Kalman filter (Jazwinski, 1970), (Särkkä and Sarmavuori, 2013). The solar radiation pressure

parameters needed to compute $\mathbf{f}$ are fixed to the values given in (Ala-Luhtala *et al*., 2012). Satellite PRN 1 was replaced since results in (Ala-Luhtala *et al*., 2012), so new values $\alpha_1 = 1.5464$ and $\alpha_2 = 0.0033$ were estimated.

We seek an estimate for the parameters $\mathbf{q}$ by maximizing the posterior distribution in Eq. (18). The prior is taken to be uniform $p(\mathbf{q}) \propto 1$. With the uniform prior, the maximum can be found by minimizing the negative log-likelihood $-\log p(\mathbf{x}_0, \dots, \mathbf{x}_N|\mathbf{q})$. Using the multivariate normal probability densities for the conditional distributions in the likelihood, the negative log-likelihood is given by

$$-\log p(\mathbf{x}_0, \dots, \mathbf{x}_N) = \frac{1}{2}\sum_{i=1}^{N} [\log|\mathbf{\Sigma}_i(\mathbf{q})| + (\mathbf{x}_i - \mathbf{m}_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{q})(\mathbf{x}_i - \mathbf{m}_i)] + \text{const.} \tag{23}$$

All the terms that do not depend on the parameter vector $\mathbf{q}$ are absorbed into the constant term. A conjugate gradient method (Luenberger and Ye, 2008) can be used for the minimization of Eq. (23). The expressions for the Jacobian and Hessian are computed analytically.

We estimate the parameters using precise ephemeris data from GPS weeks 1670-1686. For each day during the time period, we estimate a value for the $\mathbf{q}$. This provides a time series of parameter estimates. We want a single time independent value for $\mathbf{q}$, so we take the median of the resulting time series. We exclude estimates made during eclipse seasons. As an example, the median values for satellite PRN 3 are $\mathbf{q} = [-32.3 \quad -29.5 \quad -29.2]$. The values for the other satellites in the GPS constellation are $q_R \in$ [-30.4, -32.5], $q_T \in$ [-27.7, -29.9], $q_N \in$ [-28.9, -29.6].

2.4 Online estimation of initial state

Our previous studies have shown that the velocity computed directly from the satellite's broadcast ephemeris is much too inaccurate for prediction of several days (Seppänen *et al*., 2011), (Seppänen *et al*., 2012). Also, the broadcast position and velocity must be transformed from the ECEF coordinate system to the inertial coordinate system using Eq. (11). This transformation depends on the EOP values, which are not currently part of the navigation message. In addition, the satellite's solar radiation pressure parameters must also be estimated. This leads to the problem of estimating the satellite's initial state (i.e. the positon and velocity in an inertial reference frame, the EOP values and the solar radiation pressure parameters) using the data available in the navigation message. In the following, a Bayesian filtering based solution for the problem is presented.

The state-space model for the problem is given by

$$d\mathbf{x} = \mathbf{f}(\mathbf{x})dt + \mathbf{D}(\mathbf{x})d\boldsymbol{\beta}, \tag{24}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}) + \boldsymbol{\epsilon}_k, \tag{25}$$

where

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{r}, \mathbf{v}, \mathbf{p}) = \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{r}, \mathbf{p}) \\ \mathbf{0}_{d\times 1} \end{bmatrix}, \tag{26}$$

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{r}, \mathbf{v}, \mathbf{p}) = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times d} \\ \mathbf{L}(\mathbf{r}, \mathbf{v}) & \mathbf{0}_{3\times d} \\ \mathbf{0}_{\mathrm{d}\times 3} & \mathbf{I}_{d\times d} \end{bmatrix}, \quad d\boldsymbol{\beta} = \begin{bmatrix} d\boldsymbol{\beta}_{\mathbf{a}} \\ d\boldsymbol{\beta}_{\mathbf{p}} \end{bmatrix}, \tag{27}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{r}, \mathbf{v}, \mathbf{p}) = [\mathbf{T}(\mathbf{p}) \quad \mathbf{0}_{3\times 3} \quad \mathbf{0}_{3\times d}] \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{p} \end{bmatrix}. \tag{28}$$

The state of the satellite $\mathbf{x}$ consists of the position and velocity of the satellite in the inertial reference frame, and the d-dimensional ($d = 4$) vector $\mathbf{p}$, that contains the EO-parameters x_p and y_p, and the solar radiation parameters α_1 and α_2. We note, that the acceleration term $\mathbf{a}(\mathbf{r}, \mathbf{p})$ depends not only on the solar radiation pressure parameters, but also on the EO-parameters, since coordinate transformation to ECEF reference frame is needed to compute the gravitational acceleration caused by the Earth (Seppänen *et al*., 2011), (Seppänen *et al*., 2012). For the parameter vector $\mathbf{p}$ we assume a simple model

$$d\mathbf{p} = d\boldsymbol{\beta}_{\mathbf{p}}, \tag{29}$$

where $\boldsymbol{\beta}_{\mathbf{p}}$ is a d-dimensional Brownian motion stochastic process with a diagonal diffusion matrix $\mathbf{Q}_{\mathbf{p}}$. That is, we are assuming that the parameters stay approximately constant over the broadcast ephemeris's time interval. Equation (20) gives now the augmented system of equations Eq. (9) and Eq. (22). The Brownian motions $\boldsymbol{\beta}_{\mathbf{a}}$ and $\boldsymbol{\beta}_{\mathbf{p}}$ are assumed independent, so that the diffusion matrix for the joint process is

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{\mathbf{a}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\mathbf{p}} \end{bmatrix}. \tag{30}$$

The measurements $\mathbf{y}_k$ are the ECEF positions computed from the broadcast ephemeris and the matrix $\mathbf{T(p)}$ is a transformation matrix from the inertial reference frame to the ECEF reference frame. For GPS, the navigation message is valid for a 4-hour time interval from $t_{\mathrm{toe}} - 2h$ to $t_{\mathrm{toe}} + 2h$, where t_{toe} is the time of ephemeris. Using the 16 Kepler-like parameters included in the navigation message, we can compute the satellite's ECEF position at any time during the 4-hour time interval. Note that the measurement model is nonlinear, since the parameters $\mathbf{p}$ appear nonlinearly in the matrix $\mathbf{T(p)}$. The measurement noise $\boldsymbol{\epsilon}_k$ is assumed to be zero mean Gaussian white noise with covariance matrix $\mathbf{R}$. The covariance matrix $\mathbf{R}$ is chosen to be diagonal, with equal variances $r = (1\ \mathrm{m})^2$ for each coordinate axis. This is the square of the reference accuracy of GPS broadcast position (IGS, 2013).

We want to estimate the state at time t_k given all the measurements up to that time. The solution for this Bayesian filtering problem is the posterior distribution $p(\mathbf{x}(t_k)|\, \mathbf{y}_{1:k})$. The filtering algorithm recursively computes the posterior distribution, starting from a prior distribution $p(\mathbf{x}(t_0))$. In the prediction step of the filter, we compute the distribution $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k-1})$. In the update step, the information from the newest measurement is used to get the distribution $p(\mathbf{x}(t_k)|\, \mathbf{y}_{1:k})$.

In the nonlinear problem considered in this paper, the computations are intractable and approximations must be used. We consider here the Gaussian filtering approximation (Ito and Xiong, 2000), (Särkkä and Sarmavuori, 2013), where we approximate the posterior distribution with a normal distribution

$$p(\mathbf{x}(t_k) \mid \mathbf{y}_{1:k}) \approx \text{Normal}(\mathbf{x}(t_k) \mid \mathbf{m}(t_k), \mathbf{P}(t_k)). \tag{31}$$

Using this approximation, the prediction step of the filter reduces to solving the ordinary differential equations

$$\frac{d\mathbf{m}}{dt} = \mathbb{E}[\mathbf{f}(\mathbf{x})], \tag{32}$$

$$\frac{d\mathbf{P}}{dt} = \mathbb{E}[(\mathbf{x} - \mathbf{m})\mathbf{f}^{\mathbf{T}}(\mathbf{x})] + \mathbb{E}[\mathbf{f}(\mathbf{x})(\mathbf{x} - \mathbf{m})^{\mathbf{T}}] + \mathbb{E}[\mathbf{\Sigma}(\mathbf{x})], \tag{33}$$

where

$$\mathbf{\Sigma}(\mathbf{x}) = \mathbf{D}(\mathbf{x})\mathbf{Q}\mathbf{D}^{\mathbf{T}}(\mathbf{x}). \tag{34}$$

The Eqs. (32) and (33) are integrated from t_{k-1} to t_k using initial conditions $\mathbf{m}(t_{k-1})$ and $\mathbf{P}(t_{k-1})$ from the previous filtering step.

Let $\mathbf{m}^-(t_k)$ and $\mathbf{P}^-(t_k)$ be the solutions to the differential equations (32) and (33) at the end point t_k. The update step for the filter is given by

$$\boldsymbol{\mu}_k = \mathbb{E}[\mathbf{h}(\mathbf{x})] \tag{35}$$

$$\mathbf{S}_k = \mathbb{E}[(\mathbf{h}(\mathbf{x}) - \boldsymbol{\mu}_k)(\mathbf{h}(\mathbf{x}) - \boldsymbol{\mu}_k)^T] + \mathbf{R} \tag{36}$$

$$\mathbf{D}_k = \mathbb{E}[(\mathbf{x} - \mathbf{m}_k^-)(\mathbf{h}(\mathbf{x}) - \boldsymbol{\mu}_k)^T] \tag{37}$$

$$\mathbf{K}_k = \mathbf{D}_k\mathbf{S}_k^{-\mathbf{1}} \tag{38}$$

$$\mathbf{m}(t_k) = \mathbf{m}^-(t_k) + \mathbf{K}_k(\mathbf{y}_k - \boldsymbol{\mu}_k) \tag{39}$$

$$\mathbf{P}(t_k) = \mathbf{P}^-(t_k) - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \tag{40}$$

The expectations are now taken with respect to the distribution $\text{Normal}(\mathbf{m}^-(t_k), \mathbf{P}^-(t_k))$.
The Gaussian expectations in Eqs. (32), (33) and (35)-(37) are computed using numerical approximations. Using different approximations, different filters are obtained, as follows.

2.3.1 The extended Kalman filter

The extended Kalman filter can be derived by linearizing the nonlinear process and measurement functions using first order Taylor polynomial. The prediction step of the extended Kalman filter is given by (Jazwinski, 1970), (Särkkä and Sarmavuori, 2013)

$$\frac{d\mathbf{m}}{dt} = \mathbf{f}(\mathbf{m}) \tag{41}$$

$$\frac{d\mathbf{P}}{dt} = \mathbf{P}\mathbf{F}_{\mathbf{x}}^{\mathbf{T}}(\mathbf{m}) + \mathbf{F}_{\mathbf{x}}(\mathbf{m})\mathbf{P} + \mathbf{\Sigma}(\mathbf{m}) \tag{42}$$

where $\mathbf{F_x}$ is the Jacobian matrix of **f.** The update step equations (35)-(37) are given by

$$\boldsymbol{\mu}_k = \mathbf{h}(\mathbf{m}^-(t_k)) \tag{43}$$

$$\mathbf{S}_k = \mathbf{H}_{\mathbf{x}}\mathbf{P}_k^-\mathbf{H}_{\mathbf{x}}^T + \mathbf{R} \tag{44}$$

$$\mathbf{D}_k = \mathbf{P}_k^-\mathbf{H}_{\mathbf{x}}^T \tag{45}$$

where $\mathbf{H_x}$ is the Jacobian matrix of **h**, evaluated at $\mathbf{m}^-(t_k)$.

2.3.2 The unscented Kalman filter and the cubature Kalman filter

The prediction step of the UKF in the present continuous-discrete case is given by (Särkkä and Sarmavuori, 2013)

$$\frac{d\mathbf{m}}{dt} = \sum\nolimits_{i=0}^{2n} W_m^{(i)}\mathbf{f}(\mathbf{x}^{(i)}) \tag{46}$$

$$\frac{d\mathbf{P}}{dt} = \sum\nolimits_{i=0}^{2n} W_P^{(i)}\left[\mathbf{f}\left(\mathbf{x}^{(i)}\right)\boldsymbol{\xi}_i^T\sqrt{\mathbf{P}}^T + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i\mathbf{f}^T\left(\mathbf{x}^{(i)}\right) + \mathbf{\Sigma}(\mathbf{x}^{(i)})\right] \tag{47}$$

$$\mathbf{x}^{(i)} = \mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i, \tag{48}$$

$$\boldsymbol{\xi}_0 = \mathbf{0}, \qquad \boldsymbol{\xi}_i = \begin{cases} \sqrt{\lambda + n}\boldsymbol{e}_i & i = 1, \dots, n \\ -\sqrt{\lambda + n}\boldsymbol{e}_i & i = n+1, \dots, 2n \end{cases} \tag{49}$$

$$W_m^{(0)} = \frac{\lambda}{n+\lambda}, W_P^{(0)} = \frac{\lambda}{n+\lambda}(1 - \alpha^2 + \beta) \tag{50}$$

$$W_m^{(i)} = W_P^{(i)} = \frac{1}{2(n+\lambda)}, \qquad i = 1, \dots, 2n \tag{51}$$

where the matrix square root is defined to be the lower triangular matrix of the Cholesky decomposition $\mathbf{P} = \sqrt{\mathbf{P}}\sqrt{\mathbf{P}}^{\boldsymbol{T}}$ and $\lambda = \alpha^2(n + \kappa) - n$. Also, α, κ and β are parameters of the UKF. The differential equations are solved from time t_{k-1} to t_k.

Let $\mathbf{m}^-(t_k)$ and $\mathbf{P}^-(t_k)$ be the solutions at the end point. For the measurement update step, we form first the sigma-points

$$\mathcal{X}^{(i)} = \mathbf{m}^-(t_k) + \sqrt{\mathbf{P}^-(t_k)}\boldsymbol{\xi}_i, \qquad \mathcal{Y}^{(i)} = \mathbf{h}\left(\mathcal{X}^{(i)}\right), \tag{52}$$

where $\boldsymbol{\xi_i}$ are the same as in Eq. (43). Now we can approximate the expectations in Eqs. (35)-(37) with

$$\boldsymbol{\mu_k} = \sum_{i=0}^{2n} W_m^{(i)} \mathcal{Y}^{(i)} \tag{53}$$

$$\mathbf{S}_k = \sum_{i=0}^{2n} W_P^{(i)} (\mathcal{Y}^{(i)} - \boldsymbol{\mu}_k)(\mathcal{Y}^{(i)} - \boldsymbol{\mu}_k)^T \tag{54}$$

$$\mathbf{D}_k = \sum_{i=0}^{2n} W_P^{(i)} \left(\mathcal{X}^{(i)} - \mathbf{m}^-(t_k)\right)\left(\mathcal{Y}^{(i)} - \boldsymbol{\mu}_k\right)^T \tag{55}$$

Choosing $\alpha = 1$, $\beta = \kappa = 0$ we get the Cubature Kalman Filter (Särkkä and Sarmavuori, 2013).

2.5 Predicting the satellite's orbit

The orbit prediction algorithm is summarized in this section. We assume that the process noise covariance matrix has been estimated for the satellite in question. First we compute the ECEF positions, which are used as measurements, from the broadcast ephemeris. We use the time interval $t_{\text{toe}} - 1.5\text{h}$ to $t_{\text{toe}} + 1.5\text{h}$, with 5 minutes time step. The antenna offset is corrected using values provided by the NGA (NGA, 2013).

To start the filtering algorithm, we need a prior $p(\mathbf{x}(t_0))$ for the state at the initial time of $t_0 = t_{\text{toe}} - 1.5\text{h}$. We use a normal prior $p\big(\mathbf{x}(t_0)\big) = \text{Normal}(\mathbf{x}(t_0) \mid \mathbf{m}(t_0), \mathbf{P}(t_0))$, where the mean and covariance are set as follows. For the position, we take the prior mean in ECEF coordinates from the broadcast message. The variance is taken to be $(1\text{ m})^2$, which is the square of the reference accuracy for GPS broadcast position (IGS, 2013). For velocity, the prior mean is the ECEF velocity computed from the broadcast message and the variance is taken to be $(10^{-4}\text{ m/s})^2$. The formulas for computing velocity from the broadcast ephemeris are given for example in (Korvenoja and Piché, 2000). The prior mean and variance for the EO-parameters are taken to be the mean and variance of the daily precise values provided by the IERS over the years 2008--2011. The prior means for the solar radiation pressure parameters are given in (Ala-Luhtala *et al*., 2012). Satellite PRN 1 was replaced since results in (Ala-Luhtala *et al*., 2012), so new values $\alpha_1 = 1.5464$ and $\alpha_2 = 0.0033$ were estimated. A prior variance of $(10^{-6})^2$ is used for both solar radiation pressure parameters.

For the prediction algorithm, we need the position and velocity prior in the inertial reference frame. The transformation to the inertial reference frame is a nonlinear function of the position, velocity and the parameters. To get the mean and variance in the inertial reference frame, we need to compute expectations of the form

$$\mathbf{m}_0^{\text{IN}} = \mathbb{E}[\mathbf{g}\big(\mathbf{x}_0^{\text{ECEF}}\big)] \tag{56}$$

$$\mathbf{P}_0^{\text{IN}} = \mathbb{E}\left[\big(\mathbf{g}\big(\mathbf{x}_0^{\text{ECEF}}\big) - \mathbf{m}_0^{\mathbf{IN}}\big)\big(\mathbf{g}\big(\mathbf{x}_0^{\text{ECEF}}\big) - \mathbf{m}_0^{\text{IN}}\big)^T\right], \tag{57}$$

where **g** is the function that transforms the position and velocity into the inertial reference frame. The expectations in Eqs. (56) and (57) can be computed using the sigma point approximations in Eqs. (53) and (54).

After the estimation of the initial state, we can start the prediction from the time $t = t_{\mathrm{toe}} +$ 1.5h. The prediction can be computed by using the same filtering equations, but omitting the update step.

3 EVALUATING TEST RESULTS

Tests are done using broadcast ephemerides from GPS weeks 1679 to 1710. Each test consists of estimating the initial state from one broadcast ephemeris, and then predicting the orbit for a 5- day interval. For the UKF we use parameter values $\alpha = 0.001$, $\kappa = 0$ and $\beta = 0$.

The prediction errors for 5 day prediction are presented in Figure 1. The results show the combined orbit prediction errors for the whole GPS satellite constellation. For each GPS satellite 40 predictions were made using different initial times. With unhealthy satellites removed, the total number of predictions made was 1215. We note that satellite PRN 24 was unavailable for the time period used in this paper. We see that all the filtering methods give very similar means. The 95% interval for the total error is about 65m. Looking at the individual RTN error components, we see that most of the error is in the tangential direction. The values for the 95% intervals of RTN errors at day 5 of prediction are approximately 4m, 62m and 7m for the R, T and N coordinates respectively. The small radial error is a favourable result, since this component has the largest effect on the pseudorange error (Seppänen *et al*., 2012). Comparing the results to the errors using our earlier implementation (Ala-Luhtala *et al*., 2012), we can conclude that the method proposed in this paper seems to have about the same accuracy in terms of RTN errors.

The consistency of the orbit prediction is assessed by determining the proportion of cases the precise position $\mathbf{r}_{\mathrm{PE}}$ is inside the 95% probability ellipsoid defined by equation

$$(\mathbf{r} - \mathbf{r}_{\mathrm{PE}})^T \mathbf{P}^{-1} (\mathbf{r} - \mathbf{r}_{\mathrm{PE}}) \leq \beta,$$

where **r** and **P** are the predicted position and corresponding covariance matrix, and β is the value of the chi-squared inverse cumulative distribution function at point 0.95, with degrees of freedom 3. The consistency of the prediction measures how well the variance of the prediction corresponds to the realised error. The results are listed in Table 1. All methods have consistencies close to the ideal value of 0.95. This is a clear improvement over our earlier prediction algorithms**,** where consistencies of 0.25-0.40 were observed for predictions of over 3 days.

	EKF	UKF	CKF
$toe + 1.5h$	0.90	0.94	0.94
Day 1	0.97	0.99	0.98
Day 2	0.96	0.98	0.97
Day 3	0.94	0.98	0.95
Day 4	0.92	0.97	0.94
Day 5	0.91	0.97	0.93

Table 1: 95% consistencies of the predicted orbits

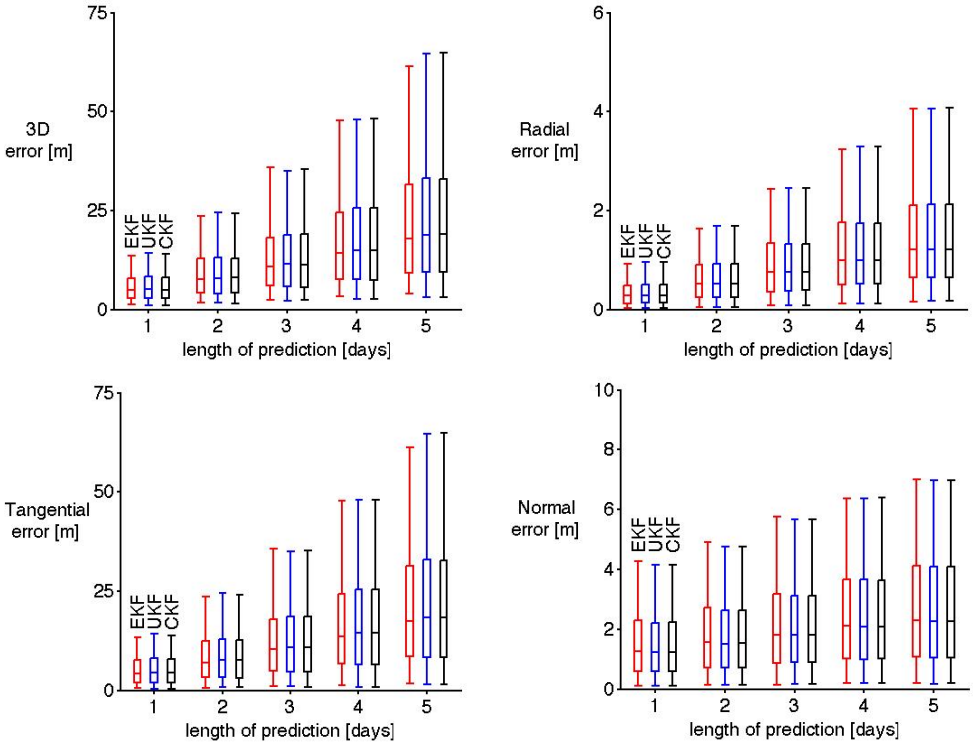


Figure 1: Box plots of the 3D and RTN errors for the different filters. The boxes present the 25%, 50% and 75% quantiles and the whiskers extending from the boxes show the 5% and 95% quantiles.

4 CONCLUSION AND DISCUSSION

This paper considers the prediction of GPS satellite orbits using information from the satellite's broadcast ephemeris. The model for the satellite's equation of motion includes the four major forces affecting the satellite: gravitational forces of Earth, Moon, and Sun, and the solar radiation pressure. The uncertainty caused by modelling errors and unmodeled forces are taken into account by including a Gaussian white noise term in the equation of motion. The covariance for the process noise is estimated using precise ephemeris data.

To start the prediction, we need to determine the satellite's initial position and velocity in an inertial reference frame, values of the EO-parameters and the solar radiation pressure parameters. We have shown how these parameters can be found using a Bayesian filtering algorithm. Three different filters were considered in this paper: the Extended Kalman filter, the Unscented Kalman filter and the Cubature Kalman filter. After the estimation of the initial state, the prediction can be carried out by computing only the prediction step of the filtering algorithm, using numerical integration to propagate the mean and variance.

The proposed method is assessed by computing the orbit prediction error in the RTN reference frame, using precise ephemerides from the IGS as reference. All the methods give

almost identical errors for the predicted orbit. Errors are largest in the tangential direction, where the 95% interval of the error is about 62m for a prediction of 5 days. The 95% intervals of the errors in the radial and normal directions are about 4 m and 7 m respectively for the 5 day prediction.

Using filtering algorithms for prediction provides an estimate for the variance of the position. We analyse the predicted variance by checking if the true position of the satellite is inside the 95% probability ellipsoid for the predicted position. The results show that all the filtering methods provide good consistencies. The UKF and CKF tend to have slightly larger values for the predicted variance and overall slightly better consistency results than the EKF.

From the results we conclude that the UKF and CKF do not seem to offer any clear improvement over the EKF. The consistency results are slightly better for the UKF and CKF, but the computational cost in our implementation is about 6 times larger than for EKF. The method proposed in this paper seems to have about the same accuracy as our previous method, where a deterministic algorithm was used to solve the initial state (Ala-Luhtala *et al*., 2012). A downside in predicting also the variance is that the differential equations are no longer independent of the velocity, and we cannot use the efficient Runge-Kutta-Nyström numerical integration method. The results for this paper were produced using Runge-Kutta method of order 4 with 15 second time step. The relatively small time step means that we need a large number of force model evaluations in the numerical integration. The computations can be made more efficient by using a more sophisticated numerical integration method, e.g. the Gauss-Jackson method (Jackson, 1924), (Berry and Healy, 2004). With the Gauss-Jackson method, we could use much larger time step and hence reduce the number of force model evaluations.

The method described here could be easily implemented also for the European Galileo and Chinese Compass satellite systems, since their broadcast ephemeris format is similar to GPS. For GLONASS the implementation is more difficult, since each broadcast ephemeris is valid only for a 30-minute time interval. This interval may be too short for accurately estimating the parameters of the model. A possible solution may be to use more than one broadcast ephemeris.

ACKNOWLEDGEMENTS

This research was funded by Nokia Inc. and Tampere Doctoral Programme in Information Science and Engineering (TISE).

REFERENCES

Ala-Luhtala J, Seppänen M, Piché R (2012) An Empirical Solar Radiation Pressure Model for Autonomous GNSS Orbit Prediction, *Proceedings of IEEE/ION The Position Location and Navigation System (PLANS) Conference 2012*, Myrtle Beach, SC, 568-575

Arasaratnam I, Haykin S (2009) Cubature Kalman filters, *IEEE Transactions on Automatic Control* 54(6):1254-1269

Berry M, Healy L (2004) Implementation of Gauss-Jackson Integration for Orbit Propagation, *The Journal of the Astronautical Sciences* 52(3): 331-357

Froideval L O (2009) *A study of solar radiation pressure acting on GPS satellites*, Ph.D. dissertation,

the University of Texas at Austin

IERS (2013) IERS Data/Products, online: http://www.iers.org/IERS/EN/DataProducts/data.html

IGS (2013) IGS Products, online: http://igscb.jpl.nasa.gov/components/prods.html

Ito K, Xiong K (2000) Gaussian Filters for Nonlinear Filtering Problems, IEEE Transactions on Automatic Control 45(5): 910-927

Jackson J (1924) Note on the Numerical Integration of $d^2x/dt^2=f(x,t)$, *Monthly Notes,* Volume 84, Royal Astronomy Society, 602-606

Jazwinski A H (1970) *Stochastic Processes and Filtering Theory*, Academic Press

Julier S J, Uhlmann J K, Durrant-Whyte H F (1995) A New Approach for Filtering Nonlinear Systems, *in the Proceedings of the 1995 American Control Conference*, vol. 3 1628-1632

Korvenoja P, Piché R (2000) Efficient satellite orbit approximation, *in Proceedings of ION GPS 2000*, 1930-1937

Luenberger D G, Ye Y (2008) *Linear and Nonlinear Programming* (3rd edition), Springer

Lytvyn M, Kemetinger A, Berglez P (2012) How can an orbit prediction module speed up the TTFF and help to authenticate the position?, *6th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing*, Noordwijk, 1-6

Mattos P G (2008) Hotstart every time – compute the ephemeris on the mobile, *Proceedings of the 21st International Technical Meeting of the Satellite Division of the Institute of Navigation*, Savannah, GA, 204-211

Montenbruck O, Gill E (2005) *Satellite Orbits* (3rd edition), Springer, 369pp

NGA (2013) NGA products, online: http://earth-info.nga.mil/GandG/saththtml/ephemeris.html

Øksendal B (2003) *Stochastic Differential Equations: An Introduction with Applications* (6th edition), Springer

Särkkä S, Sarmavuori J (2013) Gaussian filtering and smoothing for continuous-discrete dynamic systems, *Signal Processing* 93(2): 500-510

Seppänen M, Ala-Luhtala J, Piché R, Martikainen S, Ali-Löytty S (2012) Autonomous Prediction of GPS and GLONASS Satellite Orbits, *Navigation* 59(2): 119-134

Seppänen M, Perälä T, Piché R (2011) Autonomous satellite orbit prediction, *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*, San Diego, CA

Springer T, Beutler G, Rothacher M (1999) A new solar radiation pressure model for GPS satellites, *GPS Solutions*, 2:50-62

Tapley B D, Schutz B E, Born G H (2004) *Statistical Orbit Determination*, Elsevier Academic Press

Wan E A, Van der Merwe R (2000) The unscented Kalman filter for nonlinear estimation, *in the Proceedings of the Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000*, 153-158

Zhang W, Venkatasubramanian V, Liu H, Phatak M, Han S (2008) SiRF InstantFix II Technology, *Proceedings of the 21st International Technical Meeting of the Satellite Division of the Institute of Navigation*, Savannah, GA, 1840-1847

Publication 2

Juha Ala-Luhtala, Robert Piché, Gaussian scale mixture models for robust linear multivariate regression with missing data, *Communications in Statistics - Simulation and Computation*, 45:(3):791-813, 2016

Taylor & Francis
Taylor & Francis Group

Communications in Statistics - Simulation and Computation

ISSN: 0361-0918 (Print) 1532-4141 (Online) Journal homepage: http://www.tandfonline.com/loi/lssp20

Gaussian Scale Mixture Models for Robust Linear Multivariate Regression with Missing Data

Juha Ala-Luhtala & Robert Piché

To cite this article: Juha Ala-Luhtala & Robert Piché (2014): Gaussian Scale Mixture Models for Robust Linear Multivariate Regression with Missing Data, Communications in Statistics - Simulation and Computation, DOI: 10.1080/03610918.2013.875565

To link to this article: http://dx.doi.org/10.1080/03610918.2013.875565

Communications in Statistics—Simulation and Computation®, 45: 1–23, 2016

ISSN: 0361-0918 print / 1532-4141 online
DOI: 10.1080/03610918.2013.875565

Gaussian Scale Mixture Models for Robust Linear Multivariate Regression with Missing Data

JUHA ALA-LUHTALA[1] AND ROBERT PICHÉ[2]

[1]Department of Mathematics, Tampere University of Technology, Tampere, Finland
[2]Department of Automation Science and Engineering, Tampere University of Technology, Tampere, Finland

We present an algorithm for multivariate robust Bayesian linear regression with missing data. The iterative algorithm computes an approximative posterior for the model parameters based on the variational Bayes (VB) method. Compared to the EM algorithm, the VB method has the advantage that the variance for the model parameters is also computed directly by the algorithm. We consider three families of Gaussian scale mixture models for the measurements, which include as special cases the multivariate t *distribution, the multivariate Laplace distribution, and the contaminated normal model. The observations can contain missing values, assuming that the missing data mechanism can be ignored. A Matlab/Octave implementation of the algorithm is presented and applied to solve three reference examples from the literature.*

Keywords Gaussian scale mixture; Missing data; Robust linear regression; Variational Bayes

Mathematics Subject Classification Primary 62; Secondary 62J05

1. Introduction

Real datasets often contain extreme observations or outliers that are not explained by using a normal model for the observations. These outlier observations can have an unduly large influence on the inferences under the normal assumption. There is therefore interest in robust regression, robustness here meaning the tolerance of the model to outliers in the data.

Robust modeling can be based on measurement distributions having fatter tails than the normal distribution. Often used distributions in the statistical literature are Student's t distribution (Blattberg and Gonedes, 1974; West, 1984; Zellner, 1976) and the contaminated normal distribution (Huber, 1964; Tukey, 1960). A robust alternative to ordinary least squares is the method of Least Absolute Deviation (LAD) (Bloomfield and Steiger, 1983), where the absolute value of the errors is minimized instead of the squared errors. The LAD estimate is equivalent to a maximum likelihood estimate using a Laplace distribution for the measurement errors.

Received March 5, 2013; Accepted December 9, 2013
Address correspondence to Juha Ala-Luhtala, Department of Mathematics, Tampere University of Technology, PO Box 553, 33101 Tampere, Finland; E-mail: juha.ala-luhtala@tut.fi.

A common property of the three fat-tailed distributions mentioned earlier is that they can be characterized as scale mixtures of normal distributions, also called Gaussian scale mixtures (Andrews and Mallows, 1974). In the Gaussian scale mixture presentation, the measurement model is augmented with unobserved weights, so that the conditional distribution of the measurements given the parameters and the weights has a normal distribution. This kind of model enables the use of general algorithms for statistical inference.

Dempster et al. (1977) present the Expectation Maximization (EM) algorithm for Maximum Likelihood (ML) estimation in the so-called "incomplete data" models. It is also noted that the EM can be used for computing the posterior mode. Among other examples, they consider univariate linear regression with t distributed errors, which they call Iteratively Reweighted Least Squares (IRLS). An extension of IRLS for the multivariate t distribution in linear regression is presented in Rubin (2004). Little (1988) studies robust estimation of the multivariate t and contaminated normal models when the observations are allowed to contain missing values. Assuming that the missing data mechanism can be ignored (i.e., the data are missing at random (MAR)), the EM algorithm can be used to find the ML or MAP estimates of the parameters. Lange et al. (1989) consider multivariate linear and nonlinear regression using the t distribution, where the degrees of freedom are also estimated using the EM algorithm. The EM algorithm for Laplace regression is considered in Phillips (2002). Expectation Conditional Maximization (ECM) (Meng and Rubin, 1993) is an extension of the EM algorithm that simplifies the sometimes difficult implementation of the M step. The rate of convergence is improved by the extensions ECME (Liu and Rubin, 1994), Alternating Expectation Conditional Maximization (ACME) (Meng and Dyk, 1997), and Parameter Expanded Expectation Maximization (PX-EM) (Liu et al., 1998). The EM algorithm does not give directly information about the reliability of the parameter estimates. This can be addressed however by using, e.g., asymptotic results (Meng and Rubin, 1991) or bootstrapping (McLachlan and Krishnan, 2008, pp. 130–131).

In Bayesian statistical inference, we are interested in the full posterior of the model parameters. Nowadays Bayesian analysis is done mostly by Markov Chain Monte Carlo (MCMC) methods, such as the Gibbs sampler, which iteratively produce samples from the full posterior. Verdinelli and Wasserman (1991) consider Bayesian analysis of the univariate Student t and contaminated (location-shift) normal models using the Gibbs sampler. The implementation for the Student t distribution makes use of the Gaussian scale mixture presentation. Liu (1996) uses a Monte Carlo method called Data Augmentation (DA) (Tanner and Wong, 1987) for multivariate robust linear regression with missing data using the multivariate t, the contaminated normal, and the slash distribution. The algorithm makes use of the Gaussian scale mixture presentation for all the distributions. The DA and Gibbs sampler algorithms can be viewed as stochastic extensions of the EM and ECM algorithms, respectively.

An alternative for the computationally heavy Monte Carlo methods is provided by an approximate method called variational Bayes (VB). In the variational Bayesian EM (VB-EM), the intractable posterior is approximated by assuming that it factorizes between model parameters and latent variables (Beal, 2003; Beal and Ghahramani, 2003). The VB-EM algorithm iteratively minimizes the Kullback–Leibler divergence between the true posterior and the approximate distribution. The VB-EM algorithm reduces to the EM algorithm when the approximate distribution for the parameters is assumed to be a Dirac delta function (Beal and Ghahramani, 2003). Titterington (2004) provides more discussion about the VB-EM algorithm, especially in the neural networks point of view. Tipping and Lawrence (2003) use the variational approximation for robust linear interpolation using the t distribution. Penny et al. (2007) study the univariate linear regression model with

observations from the contaminated normal distribution. Wand et al. (2011) consider variational inference using the mean field method for several statistical distributions, including the Student *t* model for univariate robust regression. More examples on variational inference are provided in Ormerod and Wand (2010), where, e.g., Bayesian logistic regression using variational inference is presented. Also, it is shown how the mean field variational approximation is connected with the MCMC method Gibbs sampling. The VB approach has also been studied in connection with robust autoregressive modeling (Roberts and Penny, 2002) and nonlinear regression (Chappell et al., 2009). See also VIBES (Bishop et al., 2002), a VB-based software package for statistical inference with Bayesian networks.

The need for robust methods for statistical analysis is recognized in many statistical software packages. LIBRA (Verboven and Hubert, 2005) is a library of Matlab functions implementing many robust statistical methods, although not Bayesian regression. The `monomvn` package for R provides a Bayesian treatment of robust linear regression with missing data using the *t*-distribution; the computations are based on Gibbs sampler and DA algorithms.

Much of the literature on robust inference is concentrated on using the *t* distribution as a robust alternative for the normal distribution. Some authors also point out the use of Laplace or finite mixtures of normal distributions as other robust choices. Many of the referenced inference methods make use of the Gaussian scale mixture presentation of Andrews and Mallows (1974), especially for the *t* distribution. In this article, we present an algorithm for robust multivariate linear regression with missing data using a general Gaussian scale mixture family of distributions. The multivariate *t*, multivariate Laplace, and multivariate contaminated normal distributions are included as special cases. In the case of missing data, we assume that the missing data mechanism can be ignored as described in (Little and Rubin, 2002). The VB method provides an unified treatment of all the different models and is used to compute an approximation for the posterior distribution of the model parameters.

The rest of the paper is organized as follows. Section 2 presents the statistical model and the different distributions used for robust modeling. In Section 3, the VB method is presented and the equations needed for computing the approximate posterior are derived. The algorithm and its implementation in Matlab/Octave are also presented here. The usage of the Matlab/Octave implementation is presented in Section 4 with three examples from the literature.

2. Robust Linear Regression Model

The sampling model considered in this paper is

$$\mathbf{y}_n \mid \mathbf{x}, \mathbf{Q}, w_n \sim \text{Normal}(\mathbf{H}_n\mathbf{x}, \mathbf{Q}/w_n), \tag{1}$$

where $\mathbf{y}_n$ is a d-dimensional observation vector, $\mathbf{H}_n$ is a $d \times p$ design matrix, $\mathbf{x}$ is a p-dimensional vector of parameters, w_n is a positive scalar, and $\mathbf{Q}$ is a $d \times d$ symmetric positive definite matrix. The N observations $\mathbf{y}_1, \ldots, \mathbf{y}_N$ are assumed to be conditionally independent given the model parameters $\mathbf{x}$, $\mathbf{Q}$, w_n. The scale parameter w_n is assumed to be independent of $\mathbf{x}$, $\mathbf{Q}$.

Marginalization of w_n gives the sampling model in the form

$$\begin{aligned} p(\mathbf{y}_n \mid \mathbf{x}, \mathbf{Q}) &= \int p(\mathbf{y}_n, w_n \mid \mathbf{x}, \mathbf{Q})\, \mathrm{d}w_n = \int p(\mathbf{y}_n \mid \mathbf{x}, \mathbf{Q}, w_n) p(w_n)\, \mathrm{d}w_n \\ &= \int \text{Normal}(\mathbf{y}_n; \mathbf{H}_n\mathbf{x}, \mathbf{Q}/w_n) p(w_n)\, \mathrm{d}w_n. \end{aligned} \tag{2}$$

Sampling models obtained using Eq. (2) are called Gaussian scale mixtures (Andrews and Mallows, 1974). Robust regression can be achieved by choosing the prior distribution $p(w_n)$ such that (2) has fatter tails than the corresponding normal distribution. The mean and variance for the distribution in Eq. (2) can be found using formulas for conditional expectation and variance (Gelman et al., 2003, p. 37). The mean is given by

$$\mathbb{E}(\mathbf{y}_n \mid \mathbf{x}, \mathbf{Q}) = \mathbb{E}(\mathbb{E}(\mathbf{y} \mid \mathbf{x}, \mathbf{Q}, w_n)) = \mathbf{H}_n\mathbf{x} \tag{3}$$

and the variance is found by

$$\begin{aligned}\mathrm{Var}(\mathbf{y}_n \mid \mathbf{x}, \mathbf{Q}) &= \mathbb{E}(\mathrm{Var}(\mathbf{y} \mid \mathbf{x}, \mathbf{Q}, w_n)) + \mathrm{Var}(\mathbb{E}(\mathbf{y} \mid \mathbf{x}, \mathbf{Q}, w_n)) \\ &= \mathbb{E}\big(w_n^{-1}\big)\mathbf{Q}. \end{aligned} \tag{4}$$

We consider three families of prior distributions for w_n.

1. A gamma distribution, where the density takes the form

$$p(w_n) \propto w_n^{\alpha-1} e^{-\beta w_n}, \quad w_n > 0. \tag{5}$$

   The variance for the observation is given by

$$\mathrm{Var}(\mathbf{y}_n \mid \mathbf{x}, \mathbf{Q}) = \frac{\beta}{\alpha - 1}\mathbf{Q}. \tag{6}$$

   In the case $\alpha = \beta = \nu/2$, and the observations have a multivariate t distribution. The observations' distribution in the general case is known as the generalized t distribution of Arellano-Valle and Bolfarine (Kotz and Nadarajah, 2004, p. 94).
2. An inverse gamma distribution, with density

$$p(w_n) \propto w_n^{-\alpha-1} e^{-\beta w_n^{-1}}, \quad w_n > 0. \tag{7}$$

   The variance for the observation is given by

$$\mathrm{Var}(\mathbf{y}_n \mid \mathbf{x}, \mathbf{Q}) = \frac{\alpha}{\beta}\mathbf{Q}. \tag{8}$$

   In the case $\alpha = \beta = 1$, the random variable w_n^{-1} has a standard exponential distribution, and the observations have a multivariate symmetric Laplace distribution (Kotz et al., 2001, p. 246).
3. A two-component Gaussian mixture, with density

$$p(w_n) = (1 - \epsilon)\delta(w_n - 1) + \epsilon\delta(w_n - 1/c). \tag{9}$$

   This gives the "contaminated normal" observation model introduced by Tukey (Tukey, 1960)

$$p(\mathbf{y}_n \mid \mathbf{x}, \mathbf{Q}) = (1 - \epsilon)\mathrm{Normal}(\mathbf{y}_n; \mathbf{H}_n\mathbf{x}, \mathbf{Q}) + \epsilon\mathrm{Normal}(\mathbf{y}_n; \mathbf{H}_n\mathbf{x}, c\mathbf{Q}), \tag{10}$$

   where $0 < \epsilon < 1$ is the probability of getting an outlier and the factor $c > 1$ is used to model the larger variance of the outliers.

The method to estimate the hyperparameters, for the case that these are unknown, is given in Section 3.2. However, this generally requires a lot of data to be useful. Also, the estimation of the factor c in the contaminated normal model cannot be included in the

variational Bayes algorithm considered in this article. However, as described in Section 3.2, a grid search based method can be used instead.

The observations are allowed to contain missing data elements. We consider the case when data are missing at random (MAR) as described in Little and Rubin (2002, p. 12). Under the MAR assumption, we can ignore the missing data mechanism and base our inference solely on the observed data. For each observation $\mathbf{y}_n$, define a permutation matrix $\mathbf{M}_n = [\,\mathbf{M}_n^{\text{obs}}\ \mathbf{M}_n^{\text{miss}}\,]$ such that

$$\mathbf{y}_n = \mathbf{M}_n \begin{bmatrix} \mathbf{y}_n^{\text{obs}} \\ \mathbf{y}_n^{\text{miss}} \end{bmatrix}, \tag{11}$$

where the d_n vector $\mathbf{y}_n^{\text{obs}}$ and $d - d_n$ vector $\mathbf{y}_n^{\text{miss}}$ are the observed and missing part, respectively. If $\mathbf{y}_n$ has no missing values, take $\mathbf{M}_n = \mathbf{I}$. Data vectors with all the elements missing are discarded, since under the assumption of randomly missing data, these do not contain any useful information. The sets $\mathbf{Y}^{\text{obs}}$ and $\mathbf{Y}^{\text{miss}}$ denote all the observed and missing data, respectively.

We take independent priors for the parameters $p(\mathbf{x}, \mathbf{Q}) = p(\mathbf{x})p(\mathbf{Q})$, where

$$p(\mathbf{x}) \propto 1 \tag{12}$$

and

$$p(\mathbf{Q}) \propto |\mathbf{Q}|^{-(m+d+1)/2} \exp\left[-\tfrac{1}{2}\text{tr}(\mathbf{Q}^{-1}\mathbf{A})\right]. \tag{13}$$

The distribution $p(\mathbf{Q})$ is the Inverse Wishart distribution, where m and $\mathbf{A}$ are parameters. Taking $m = 0$ and $\mathbf{A} = 0$, we get the noninformative Jeffreys prior (Gelman et al., 2003).

3. Approximate Bayesian Inference

3.1. *Theory*

The solution of the linear regression problem is the posterior distribution $p(\mathbf{x}, \mathbf{Q} \mid \mathbf{Y}^{\text{obs}})$. This distribution can be obtained by integrating out the latent variables and the missing data from the full posterior

$$p\big(\mathbf{x}, \mathbf{Q} \,\big|\, \mathbf{Y}^{\text{obs}}\big) = \int\int p\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}} \,\big|\, \mathbf{Y}^{\text{obs}}\big)\, \mathrm{d}\mathbf{w}\ \mathrm{d}\mathbf{Y}^{\text{miss}}. \tag{14}$$

In this article, we consider a Variational Bayes (VB) approximation to the posterior distribution. The variational approach in general is based on maximizing the variational lower bound of the logarithm of the marginal likelihood (Beal, 2003; Beal and Ghahramani, 2003; Bishop, 2006; MacKay, 2003)

$$\log p\big(\mathbf{Y}^{\text{obs}}\big) \geq \int q\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}\big) \log \frac{p\big(\mathbf{Y}^{\text{obs}}, \mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}\big)}{q\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}\big)}\, \mathrm{d}\mathbf{x}\ \mathrm{d}\mathbf{Q}\ \mathrm{d}\mathbf{w}\ \mathrm{d}\mathbf{Y}^{\text{miss}}, \tag{15}$$

where $q(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}})$ is any distribution over the latent variables and model parameters. It can be shown that maximizing the lower bound is equivalent to minimizing the

Kullback–Leibler divergence between $q(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}})$ and the full posterior:

$$\text{KL}(q \parallel p) = \int q\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}\big) \log \frac{q\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}\big)}{p\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}} \,\big|\, \mathbf{Y}^{\text{obs}}\big)} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{Q} \, \mathrm{d}\mathbf{w} \, \mathrm{d}\mathbf{Y}^{\text{miss}}. \quad (16)$$

In the VB approach we make a fully factorized approximation for the posterior distribution (Bishop, 2006; MacKay, 2003)

$$\begin{aligned} p\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}} \,\big|\, \mathbf{Y}^{\text{obs}}\big) &\approx q(\mathbf{x}) q(\mathbf{Q}) q(\mathbf{w}) q\big(\mathbf{Y}^{\text{miss}}\big) \\ &= q(\mathbf{x}) q(\mathbf{Q}) \prod_{n=1}^{N} q(w_n) q\big(\mathbf{y}_n^{\text{miss}}\big). \end{aligned} \quad (17)$$

That is, the posterior model parameters and latent variables are approximated as being mutually independent. Note that the functional form of the approximating distributions $q(\cdot)$ is not fixed. Generally, the approximations from variational inference tend to be more compact than the true distribution (MacKay, 2003, p. 431). It can be shown (Bishop, 2006) that the optimal distributions in the sense of KL-divergence satisfy the equations

$$\log q(s) = \mathbb{E}_{-s} \log p\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}, \mathbf{Y}^{\text{obs}}\big), \quad (18)$$

where

$$s \in \big\{\mathbf{x}, \mathbf{Q}, w_1, \ldots, w_N, \mathbf{y}_1^{\text{miss}}, \ldots, \mathbf{y}_N^{\text{miss}}\big\}. \quad (19)$$

The notation $\mathbb{E}_{-s}(\cdot)$ means that the expectation is taken with respect to all variables other than s. The parameters of the approximation can be found by fixed-point iteration based on Eq. (18).

We next derive the equations for computing the approximating distributions using Eq. (18). The log-probability of the joint distribution of all the variables is

$$\begin{aligned} &\log p\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}, \mathbf{Y}^{\text{obs}}\big) \\ &\quad = -\frac{N + m + d + 1}{2} \log |\mathbf{Q}| - \frac{1}{2} \text{tr}(\mathbf{Q}^{-1} \mathbf{A}) \\ &\qquad \times \sum_{n=1}^{N} \left[-\frac{w_n}{2} (\mathbf{y}_n - \mathbf{H}_n \mathbf{x})^T \mathbf{Q}^{-1} (\mathbf{y}_n - \mathbf{H}_n \mathbf{x}) + \frac{d}{2} \log w_n + \log p(w_n) \right] + \text{const}, \end{aligned} \quad (20)$$

where $\mathbf{y}_n$ is of the form in Eq. (11). Taking expectation with respect to all other variables than $\mathbf{x}$ and absorbing all the terms that do not involve $\mathbf{x}$ into the constant term, we get

$$\log q(\mathbf{x}) = \sum_{n=1}^{N} -\frac{\bar{w}_n}{2} \left[\mathbf{x}^T \mathbf{H}_n^T \mathbf{S} \mathbf{H}_n \mathbf{x} - 2 \mathbf{x}^T \mathbf{H}_n^T \mathbf{S} \bar{\mathbf{y}}_n \big) \right] + \text{const}, \quad (21)$$

where $\bar{w}_n = \mathbb{E}(w_n)$, $\mathbf{S} = \mathbb{E}(\mathbf{Q}^{-1})$ and

$$\bar{\mathbf{y}}_n = \mathbf{M}_n \begin{bmatrix} \mathbf{y}_n^{\text{obs}} \\ \mathbb{E}\big(\mathbf{y}_n^{\text{miss}}\big) \end{bmatrix}. \quad (22)$$

The expression in Eq. (21) is recognized as the logarithm of the normal density with variance

$$\mathbf{P} = \left(\sum_{n=1}^{N} \bar{w}_n \mathbf{H}_n^T \mathbf{S} \mathbf{H}_n \right)^{-1} \tag{23}$$

and mean

$$\bar{\mathbf{x}} = \mathbf{P} \left(\sum_{n=1}^{N} \bar{w}_n \mathbf{H}_n^T \mathbf{S} \bar{\mathbf{y}}_n \right). \tag{24}$$

To find the distribution for $\mathbf{Q}$, we first note that the quadratic form in Eq. (48) can be written using the matrix trace operator as

$$(\mathbf{y}_n - \mathbf{H}_n \mathbf{x})^T \mathbf{Q}^{-1} (\mathbf{y}_n - \mathbf{H}_n \mathbf{x}) = \operatorname{tr}\left[\mathbf{Q}^{-1} (\mathbf{y}_n - \mathbf{H}_n \mathbf{x})(\mathbf{y}_n - \mathbf{H}_n \mathbf{x})^T \right]. \tag{25}$$

Using this result and taking again expectation with respect to all other variables than $\mathbf{Q}$, we get

$$\log q(\mathbf{Q}) = -\frac{1}{2} \operatorname{tr}\left[\mathbf{Q}^{-1} (\mathbf{A} + \mathbf{R}) \right] - \frac{N + m + d + 1}{2} \log |\mathbf{Q}| + \text{const}, \tag{26}$$

where

$$\mathbf{R} = \sum_{n=1}^{N} \bar{w}_n \left[(\bar{\mathbf{y}}_n - \mathbf{H}_n \bar{\mathbf{x}})(\bar{\mathbf{y}}_n - \mathbf{H}_n \bar{\mathbf{x}})^T + \Sigma_n + \mathbf{H}_n \mathbf{P} \mathbf{H}_n^T \right] \tag{27}$$

and

$$\Sigma_n = \mathbf{M}_n \begin{bmatrix} \underline{0} & \underline{0} \\ \underline{0} & \operatorname{Cov}\left(\mathbf{y}_n^{\text{miss}}\right) \end{bmatrix} \mathbf{M}_n^T. \tag{28}$$

This is recognized as the inverse Wishart distribution with degrees of freedom $N + m$ and scale matrix $\mathbf{A} + \mathbf{R}$. From this, it follows that $\mathbf{Q}^{-1}$ has Wishart distribution with the same degrees of freedom and scale matrix $(\mathbf{A} + \mathbf{R})^{-1}$. Using this result, we can compute

$$\mathbf{S} = \mathbb{E}(\mathbf{Q}^{-1}) = (N + m)(\mathbf{A} + \mathbf{R})^{-1}. \tag{29}$$

For one-dimensional measurements (i.e., $d = 1$), the inverse Wishart distribution reduces to the inverse Gamma distribution.

For the missing values, we find that

$$\log q\left(\mathbf{y}_n^{\text{miss}}\right) = -\frac{\bar{w}_n}{2} \left[\mathbf{y}_n^T \mathbf{S} \mathbf{y}_n - 2 \mathbf{y}_n^T \mathbf{S} \mathbf{H}_n \bar{\mathbf{x}} \right] + \text{const}. \tag{30}$$

Inserting Eq. (11) for $\mathbf{y}_n$, we find that the distributions for the missing values are normal with means

$$\mathbb{E}\left(\mathbf{y}_n^{\text{miss}}\right) = \left(\mathbf{M}_n^{\text{miss}}\right)^T \mathbf{H}_n \bar{\mathbf{x}} + \left(\Sigma_n^{\text{om}}\right)^T \left(\Sigma_n^{\text{obs}}\right)^{-1} \left(\mathbf{y}_n^{\text{obs}} - \mathbf{M}_n^{\text{obs}} \mathbf{H}_n \bar{\mathbf{x}}\right) \tag{31}$$

and variance

$$\operatorname{Cov}\left(\mathbf{y}_n^{\text{miss}}\right) = \Sigma_n^{\text{miss}} - \left(\Sigma_n^{\text{om}}\right)^T \left(\Sigma_n^{\text{obs}}\right)^{-1} \Sigma_n^{\text{om}}, \tag{32}$$

where

$$\Sigma_n^{\text{obs}} = \left(\mathbf{M}_n^{\text{obs}}\right)^T [\bar{w}_n \mathbf{S}]^{-1} \mathbf{M}_n^{\text{obs}}, \tag{33}$$

$$\Sigma_n^{\text{miss}} = \left(\mathbf{M}_n^{\text{miss}}\right)^T [\bar{w}_n \mathbf{S}]^{-1} \mathbf{M}_n^{\text{miss}}, \tag{34}$$

$$\Sigma_n^{\text{om}} = \left(\mathbf{M}_n^{\text{obs}}\right)^T [\bar{w}_n \mathbf{S}]^{-1} \mathbf{M}_n^{\text{miss}}. \tag{35}$$

The form of the optimal distribution for weights w_n depends on the prior distribution $p(w_n)$

$$q(w_n) \propto w_n^{d/2} \exp\left(-\frac{w_n}{2} l_n\right) p(w_n), \tag{36}$$

where

$$l_n = (\bar{\mathbf{y}}_n - \mathbf{H}_n \mathbf{m})^T \mathbf{S} (\bar{\mathbf{y}}_n - \mathbf{H}_n \mathbf{m}) + \text{tr}\left[\mathbf{S}\left(\mathbf{H}_n \mathbf{P} \mathbf{H}_n^T + \Sigma_n\right)\right]. \tag{37}$$

The distributions for the three families of priors considered in this work are as follows:

1. For a-priori gamma-distributed w_n, the optimal distribution is also gamma distribution with density

$$q(w_n) \propto w_n^{\alpha + d/2 - 1} e^{-w_n(\beta + l_n/2)} \tag{38}$$

and mean

$$\bar{w}_n = \frac{\alpha + d/2}{\beta + l_n/2}. \tag{39}$$

2. For a priori inverse gamma-distributed w_n, the optimal distribution is

$$q(w_n) \propto w_n^{d/2 - \alpha - 1} e^{-\frac{1}{2}(l_n w_n + 2\beta w_n^{-1})}. \tag{40}$$

This is recognized as a Generalized Inverse Gaussian (GIG) distribution (Jørgensen, 1982). The GIG distribution has mean

$$\bar{w}_n = \sqrt{\frac{2\beta}{l_n}} \frac{K_{d/2-\alpha+1}(\sqrt{2\beta l_n})}{K_{d/2-\alpha}(\sqrt{2\beta l_n})}, \tag{41}$$

where K_p is the modified Bessel function of the second kind with order p.

3. For the discrete distribution of Eq. (9), the optimal distribution is the discrete distribution

$$q(w_n) \propto (1-\epsilon) e^{-\frac{l_n}{2}} \delta(w_n - 1) + \epsilon c^{-d/2} e^{-\frac{l_n}{2c}} \delta(w_n - 1/c), \tag{42}$$

with mean

$$\bar{w}_n = \frac{(1-\epsilon) e^{-\frac{l_n}{2}} + \epsilon c^{-d/2-1} e^{-\frac{l_n}{2c}}}{(1-\epsilon) e^{-\frac{l_n}{2}} + \epsilon c^{-d/2} e^{-\frac{l_n}{2c}}}. \tag{43}$$

3.1.1. *Computation of the Variational Lower Bound.* The lower bound of the marginal likelihood defined in Eq. (15) should be nondecreasing during the iterative variational inference algorithm (Bishop, 2006, p. 481) and can be used to check the convergence of the algorithm. Expanding the expression for the lower bound, we have

$$L(q) = \mathbb{E}\log p\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}, \mathbf{Y}^{\text{obs}}\big) - \mathbb{E}\log q(\mathbf{x}) - \mathbb{E}\log q(\mathbf{Q}) + \sum_{n=1}^{N} -\mathbb{E}\log q(w_n) + \sum_{n=1}^{N} -\mathbb{E}\log q\big(\mathbf{y}_n^{\text{miss}}\big), \tag{44}$$

where the first expectation is with respect to the whole approximate posterior in Eq. (17) and the rest with respect to the corresponding approximate marginal distributions. The terms of the form $-\mathbb{E}\log q(s)$ are recognized as the differential entropies of the corresponding distributions. The entropies for $q(\mathbf{x})$, $q(\mathbf{Q})$, and $q(\mathbf{y}_n^{\text{miss}})$ are computed using equations (Bishop, 2006, pp. 685–693)

$$-\mathbb{E}\log q(\mathbf{x}) = \frac{1}{2}\log|\mathbf{P}| + \frac{p}{2}[1 + \log(2\pi)] \tag{45}$$

$$-\mathbb{E}\log q(\mathbf{Q}) = \frac{d+1}{2}\log|\mathbf{A}+\mathbf{R}| + \frac{d(N+m)}{2} + \log\left[2^{d(N+m)/2}\pi^{d(d-1)/4}\prod_{j=1}^{d}\Gamma\left(\frac{N+m+(1-j)}{2}\right)\right] \tag{46}$$

$$-\mathbb{E}\log q\big(\mathbf{y}^{\text{miss}}\big) = \frac{1}{2}\log\big|\text{Cov}\big(\mathbf{y}_n^{\text{miss}}\big)\big| + \frac{d_n}{2}[1 + \log(2\pi)]. \tag{47}$$

The expression for $\mathbf{P}$, $\mathbf{R}$, and $\text{Cov}(\mathbf{y}_n^{\text{miss}})$ are given by Eqs. (23), (27), and (32), respectively. The entropy of the inverse Wishart distribution can be derived using entropy of the Wishart distribution.

Inserting the expression in Eq. (20) for the first term in the lower bound and evaluating the expectations, we get

$$\mathbb{E}\log p\big(\mathbf{x}, \mathbf{Q}, \mathbf{w}, \mathbf{Y}^{\text{miss}}, \mathbf{Y}^{\text{obs}}\big) = -\frac{N+m+d+1}{2}\log|\mathbf{A}+\mathbf{R}| - \frac{1}{2}\text{tr}(\mathbf{SA}) + \sum_{n=1}^{N}\left[-\frac{\bar{w}_n}{2}l_n + \frac{d}{2}\mathbb{E}\log w_n + \mathbb{E}\log p(w_n)\right] + \text{const}, \tag{48}$$

where the terms $\mathbf{R}$, $\mathbf{S}$, and l_n are given by Eqs. (27), (29), and (37), respectively. The constant term in the expression, i.e., the term that does not change during iterations, is given by

$$\text{const} = \frac{m}{2}\log|\mathbf{A}| - \log\left[2^{md/2}\pi^{d(d-1)/4}\prod_{j=1}^{d}\Gamma\left(\frac{m+1-j}{2}\right)\right] - \frac{d}{2}\log(2\pi). \tag{49}$$

The expressions for the terms depending on the prior distribution of w_n are given as follows:

1. For a-priori gamma-distributed w_n, the optimal distribution is the gamma distribution $w_n \sim \text{Gamma}(a_n, b_n)$, with parameters $a_n = \alpha + d/2$ and $b_n = \beta + l_n/2$. The mean $\bar{w}_n$ is given by Eq. (39). The log-expectation and the entropy are given by (Bishop, 2006, p. 688)

$$\mathbb{E} \log w_n = \psi(a_n) - \log b_n \tag{50}$$

and

$$-\mathbb{E} \log q(w_n) = \log \Gamma(a_n) - (a_n - 1)\psi(a_n) - \log b_n + a_n, . \tag{51}$$

where $\psi(x)$ is the digamma function. The expectation of the prior log-probability is given by

$$\mathbb{E} \log p(w_n) = \alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1)\mathbb{E} \log w_n - \beta \bar{w}_n. \tag{52}$$

2. For a-priori inverse gamma-distributed w_n, the optimal distribution is the GIG distribution $w_n \sim \text{GIG}(p_n, a_n, b_n)$, with parameters $p_n = d/2 - \alpha$, $a_n = l_n$ and $b_n = 2\beta$. The mean $\bar{w}_n$ is given by Eq. (41). The log-expectation is given by (Jørgensen, 1982, p. 21)

$$\mathbb{E} \log w_n = \log \left(\sqrt{\frac{b_n}{a_n}} \right) + \frac{\left[\frac{\partial}{\partial v} K_v(\sqrt{a_n b_n}) \right]_{v=p_n}}{K_p(\sqrt{a_n b_n})}. \tag{53}$$

The partial derivative of the modified Bessel function with respect to the order v can be evaluated using formulas from Abramowitz and Stegun (1965, p. 377). The entropy is given by

$$\begin{aligned} -\mathbb{E} \log q(w_n) = \log \left[\frac{(a_n/b_n)^{p_n/2}}{2K_{p_n}(\sqrt{a_n b_n})} \right] + (p-1)\mathbb{E} \log w_n \\ - \frac{1}{2} \left(a_n \bar{w}_n + b_n \mathbb{E} w_n^{-1} \right), \end{aligned} \tag{54}$$

where

$$\mathbb{E} w_n^{-1} = \sqrt{\frac{a_n}{b_n}} \frac{K_{p-1}(\sqrt{a_n b_n})}{K_{p_n}(\sqrt{a_n b_n})}. \tag{55}$$

The expectation of the prior log-probability is given by

$$\mathbb{E} \log p(w_n) = \alpha \log \beta - \log \Gamma(\alpha) - (\alpha + 1)\mathbb{E} \log w_n - \beta \mathbb{E} w_n^{-1}. \tag{56}$$

3. For the discrete distribution of Eq. (42) the mean $\bar{w}_n$ is given by Eq. (43). The log-expectation and entropy are given by

$$\mathbb{E} \log w_n = \log \left(\frac{1}{c} \right) q(w_n = 1/c) \tag{57}$$

and

$$-\mathbb{E} \log q(w_n) = -q(w_n = 1) \log q(w_n = 1) - q(w_n = 1/c) \log q(w_n = 1/c), \tag{58}$$

where $q(w_n = s)$ is the value of the pdf $q(w_n)$ evaluated at point s. The expectation of the prior log-probability is given by

$$\mathbb{E}\log p(w_n) = q(w_n = 1)\log(1-\epsilon) + q(w_n = 1/c)\log(\epsilon). \tag{59}$$

Collecting the results, the expression for the lower bound is given by

$$L(q) = -\frac{N+m}{2}\log|\mathbf{A}+\mathbf{R}| + \frac{1}{2}\log|\mathbf{P}| + \frac{1}{2}\sum_{n=1}^{N}\log\left|\mathrm{Cov}\left(\mathbf{y}_n^{\mathrm{miss}}\right)\right| - \frac{1}{2}\mathrm{tr}(\mathbf{SA})$$
$$+ \sum_{n=1}^{N}\left[-\frac{\bar{w}_n}{2}l_n + \frac{d}{2}\mathbb{E}\log w_n + \mathbb{E}\log p(w_n)\right] + \mathrm{const}, \tag{60}$$

where $\bar{w}_n$, $\mathbb{E}\log w_n$ and $\mathbb{E}\log p(w_n)$ are computed using results depending on the used prior density $p(w_n)$, and the constant term is given by

$$\mathrm{const} = \frac{p + d(N+m) + \sum_{n=1}^{N} d_n)}{2} + \log(2\pi)\frac{p - d + \sum_{n=1}^{N} d_n}{2}$$
$$+ \log\left[2^{dN/2}\prod_{j=1}^{d}\frac{\Gamma\left(\dfrac{N+m+(1-j)}{2}\right)}{\Gamma\left(\dfrac{m+1-j}{2}\right)}\right] + \frac{m}{2}\log|\mathbf{A}|. \tag{61}$$

3.1.2. *Estimating the Hyperparameters.* The hyperparameters of the prior distribution for the weights $p(w_n)$ might be unknown. To estimate also the hyperparameters, an additional maximization step can be included to the variational Bayes algorithm as described by Beal (Beal, 2003, pp. 61–62). The iteration for the hyperparameter estimation proceeds as follows. Given the previous estimates for the hyperparameters, the optimal distributions for the variables $\mathbf{x}$, $\mathbf{Q}$, $\mathbf{w}$, and $\mathbf{Y}^{\mathrm{miss}}$ are computed. After this, the obtained optimal distributions are kept unchanged and the lower bound is maximized with respect to the hyperparameters. For the Gaussian scale mixture models used in this article, the maximization of the lower bound with respect to the hyperparameters reduces to maximizing the sum of the expected log-probabilites of the prior distribution:

$$\sum_{n=1}^{N}\mathbb{E}\log p(w_n). \tag{62}$$

For the three different families of prior distributions $p(w_n)$, the maximization proceeds as follows:

1. For a-priori gamma-distributed w_n, Eq. (52) is maximized with respect to the hyperparameters α and β. The terms $\mathbb{E}\log w_n$ and $\bar{w}_n$ are given by Eqs. (50) and (39), where the previous estimated values are used for the hyperparameters α and β. Taking derivatives with respect to α and β, and setting them to zero, we get the equations

$$N\log\beta - N\Psi(\alpha) + \sum_{n=1}^{N}\mathbb{E}\log(w_n) = 0 \tag{63}$$

$$N\frac{\alpha}{\beta} - \sum_{n=1}^{N} \bar{w}_n = 0, \tag{64}$$

where

$$\Psi(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} \tag{65}$$

is the digamma function. There is no explicit solution, but an iterative method (e.g., the Newton method) can be used to solve the equations. The Hessian is given by

$$\mathbf{H} = \begin{bmatrix} -N\Psi'(\alpha) & \frac{N}{\beta} \\ \frac{N}{\beta} & -N\frac{\alpha}{\beta}^2 \end{bmatrix}. \tag{66}$$

The derivative of the digamma function $\Psi'(\alpha)$ is also known as the trigamma function. The Hessian is always negative definite for feasible values of α and β (i.e. $\alpha > 0$, $\beta > 0$), so that the solution is a local maximum.

2. For a-priori inverse gamma-distributed w_n, Eq. (56) is maximized with respect to the hyperparameters α and β. The terms $\mathbb{E}\log w_n$ and $\mathbb{E}w_n^{-1}$ are computed using Eqs. (53) and (55), where values of the previous estimates are used for α and β. Taking derivatives with respect to α and β and evaluating to zero gives equations

$$N\log\beta - N\Psi(\alpha) - \sum_{n=1}^{N} \mathbb{E}\log(w_n) = 0 \tag{67}$$

$$N\frac{\alpha}{\beta} - \sum_{n=1}^{N} \mathbb{E}w_n^{-1} = 0. \tag{68}$$

The iterative method can be used to solve the obtained equations. The Hessian is the same as for the a-priori gamma distributed w_n and is always negative definite for $\alpha > 0$ and $\beta > 0$.

3. For the discrete distribution of Eq. (42), the hyperparameters are the probability of the outlier ϵ and the scaling factor c. Derivating Eq. (59) with respect to ϵ and setting the derivative to zero, gives

$$\epsilon = \frac{q(w_n = 1)}{q(w_n = 1) + q(w_n = 1/c)}. \tag{69}$$

Note that this method cannot be used to estimate the scaling factor, since c appears only as the argument of the discrete distribution q. The scaling factor c can be estimated using a grid search-based method, where the VBEM algorithm is run for several different values of c and the best estimate is chosen to be the value of c for which the final value of the lower bound is largest.

3.2. *Algorithm*

The VB algorithm for the robust regression proceeds iteratively starting from some initial guess for the statistics $\bar{\mathbf{x}}$, $\mathbf{P}$, $\mathbf{S}$, and $\bar{w}_1, \ldots, \bar{w}_N$. We first update $\bar{\mathbf{y}}_n$ and Σ_n for $n = 1, \ldots, N$

using Eqs. (31), (22), (32), and (28). Next, we use these values and update $\bar{\mathbf{x}}$, $\mathbf{P}$, and $\mathbf{S}$ using Eqs. (24), (23), (27), and (29). Last, we update the means for the weights w_n using the appropriate Eq. (39), (41), or (43). If the hyperparameters are to be estimated, an additional maximization step is included depending on the prior distribution $p(w_n)$. The iteration is repeated until convergence is achieved.

During each iteration, we check the convergence with

$$L(q_k) - L(q_{k-1}) < \text{lTol}, \tag{70}$$

In our implementation, the default value is $\text{lTol} = 10^{-8}$. We also monitor the convergence of the algorithm by comparing the absolute and relative change of the estimate of $\mathbf{x}$ in the iteration k to the estimate in the previous iteration $k-1$. The stopping criteria for the algorithm in the iteration k is

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_\infty < \max\{\text{absTol}, \text{relTol} \cdot \|\mathbf{x}_k\|_\infty\}. \tag{71}$$

Our default values are $\text{abstol} = 10^{-8}$ and $\text{reltol} = 10^{-8}$. Also, the lower bound given in Eq. (60) can be used to monitor the convergence. The pseudocode for the algorithm is given in Algorithm 1.

Algorithm 1 VB algorithm for robust linear regression

Initialize: $\bar{\mathbf{x}} \leftarrow \mathbf{0}$, $\mathbf{S} \leftarrow \mathbf{I}$, $\mathbf{P} \leftarrow \mathbf{I}$, $\bar{\mathbf{w}} \leftarrow 1$
  while do
    for $n = 1$ **to** N **do**
      Update $\bar{\mathbf{y}}_n$ using (31), (22), (32)
      Update $\boldsymbol{\Sigma}_n$ using (28)
    end for
    $\mathbf{P} \leftarrow \left(\sum_{n=1}^{N} \bar{w}_n \mathbf{H}_n^T \mathbf{S} \mathbf{H}_n\right)^{-1}$
    $\bar{\mathbf{x}} \leftarrow \mathbf{P}\left(\sum_{n=1}^{N} \bar{w}_n \mathbf{H}_n^T \mathbf{S} \bar{\mathbf{y}}_n\right)$
    $\mathbf{R} \leftarrow \sum_{n=1}^{N} \bar{w}_n \left((\bar{\mathbf{y}}_n - \mathbf{H}_n \bar{\mathbf{x}})(\bar{\mathbf{y}}_n - \mathbf{H}_n \bar{\mathbf{x}})^T + \boldsymbol{\Sigma}_n + \mathbf{H}_n \mathbf{P} \mathbf{H}_n^T\right)$
    $\mathbf{S} \leftarrow (N + m)(\mathbf{A} + \mathbf{R})^{-1}$
    for $n = 1$ **to** N **do**
      Update $\bar{w}_n$ using (39), (41) or (43)
    end for
    if Hyperparameters unknown **then**
      if $p(w_n)$ is gamma distribution **then**
        Update α and β by maximizing (63) and (64)
      else if $p(w_n)$ is inverse-gamma distribution **then**
        Update α and β by maximizing (67) and (68)
      else if $p(w_n)$ is contaminated normal distribution **then**
        Update ϵ using (69)
      end if
    end if
    Update lower bound using (60)
    Check convergence using (70), (71)
  end while

The algorithm is implemented in the Matlab function `rmvregress`, which also works in Octave. The default measurement noise distribution is the multivariate t with four degrees of freedom. The function can be freely downloaded from the Matlab Central file exchange.

4. Examples

4.1. *Stack Loss Data, Univariate Observations*

The stack loss dataset has been analyzed in the literature, for example Lange et al. (1989) and Hoeting et al. (1996). The dataset contains univariate observations of stack loss, which is assumed to depend linearly on three regressors: air flow, temperature, and acid content. Also, an intercept term is estimated.

We consider robust regression with three different noise models: Student t, Laplace, and contaminated normal distributions. We consider Student t distributions with degrees of freedom $\nu = 4$ and $\nu = 1.1$. The value $\nu = 4$ is a "general-purpose" choice, while $\nu = 1.1$ is the ML estimate for the degrees of freedom obtained in Lange et al. (1989). The prior distribution $p(\mathbf{x}, Q)$ is taken to be the noninformative Jeffreys prior. The observed data are collected to a 21-element column vector Y. The predictor variables are collected into a 21×4 matrix H, where each row represents the values for the corresponding observation in Y. The first element of each row is a 1, corresponding to the intercept term.

Regression with Student's t-distribution with $\nu = 4$ degrees of freedom is the default option, so we use the command

```
>> [x,s,W]=rmvregress(H,Y);
```

For the degrees of freedom $\nu = 1.1$, use:

```
>> [x,s,W]=rmvregress(H,Y, 'student', 1.1);
```

The Laplace regression is obtained with

```
>> [x,s,W]=rmvregress(H,Y, 'laplace');
```

For the contaminated normal, we take $\epsilon = 0.1$ and $c = 10$:

```
>> [x,s,W]=rmvregress(H,Y, 'contnorm', [0.1,10]);
```

The values of the lower bound during the VB updates are plotted in Fig. 1. As can be seen from the plot, the VB updates converge rather quickly close to the maximal value of the lower bound.

Small values in the estimated weights W correspond to possible outliers in the data. The weights for different observation distributions are collected in Table 1. As stated for example in Lange et al. (1989) and Hoeting et al. (1996), the observations 1, 3, 4. and 21 are generally considered outliers, so these rows are shaded in the table. We see that the Student t and Laplace distributions assign small weights to these four observations. The contaminated normal distribution clearly distinguishes the observations 4 and 2, but observations 1 and 3 do not stand out from the other observations.

Instead of just point estimates, we may be interested in the full posterior distribution of the parameters. The covariance matrix for the regression coefficients is obtained by including P in the output argument list:

```
>> [x,s,W, P]=rmvregress(H,Y);
```

The standard errors for the regression coefficients can be obtained by taking the square root of the diagonal elements of P. The standard errors for the different regression models are listed in Table 2. We see that the standard errors for the $t_{1.1}$ distribution are close to the asymptotic results obtained with the expected information matrix in Lange et al. (1989).

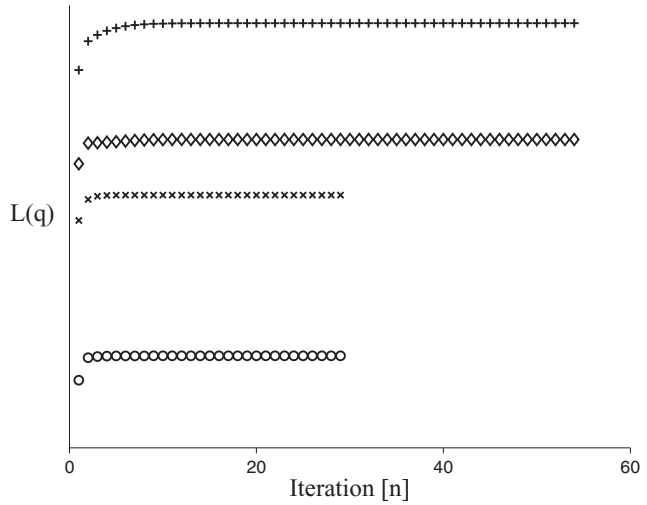


Figure 1. Computed values of the lower bound during the VB algorithm for t distribution ($\nu = 4$) (circle), t distribution ($\nu = 1.1$) (plus), Laplace distribution (cross), and contaminated normal distribution (diamond).

4.2. *Astronomy Data, Bivariate Observations*

We use the star cluster dataset of Rousseeuw and Leroy (2003) to illustrate robust estimation of the mean and the covariance matrix. The astronomy data are bivariate ($d = 2$), consisting of logarithms of the effective temperature at the surface of the star and of the light intensity of the star. We fit a multivariate t-distribution with $\nu = 5$ degrees of freedom, a multivariate Laplace distribution, and a contaminated normal distribution to the data. The prior distribution $p(\mathbf{x}, \mathbf{Q})$ is taken to be the noninformative Jeffreys prior.

The data are collected in the 47×2 matrix $\mathbf{Y}$, where each row represents one observation. The design matrix is $\mathbf{H}_n = \mathbf{I}$ for all the observations. The regression is then performed by the command

```
>> [x,Q,W,P,R,V]=rmvregress(H,Y, 'student', 5);
```

The regression with the Laplace distribution is computed using the command

```
>> [x,Q,W,P,R,V]=rmvregress(H,Y, 'laplace');
```

For the contaminated normal distribution, we take $\epsilon = 0.1$ and $c = 10$:

```
>> [x,Q,W,P,R,V]=rmvregress(H,Y, 'contnorm', [0.1, 10]);
```

The values of the lower bound during the VB updates are plotted in Fig. 2.

The obtained mean estimates for the location and the concentration ellipses, defined as (Anderson, 2003)

$$(\mathbf{x} - \mathbf{m})^T \mathbf{Q}^{-1} (\mathbf{x} - \mathbf{m}) = d + 2,$$

are plotted in Fig. 3. In the figure, the extreme observations 7, 11, 20, 30, and 34 are labeled. The weights for these observations are listed in Table 3. The rest of the weights are in the

Table 1
Expected weights w_n for the observations

Observation	t_4	$t_{1.1}$	Laplace	Contam.
1	0.80	0.11	0.98	0.94
2	1.02	1.27	3.44	0.94
3	0.68	0.10	0.88	0.90
4	0.42	0.05	0.63	0.37
5	1.12	1.23	3.63	0.96
6	1.00	0.85	2.40	0.95
7	1.09	1.45	3.78	0.96
8	1.18	1.46	5.79	0.97
9	1.04	1.08	2.78	0.96
10	1.19	1.63	5.99	0.97
11	1.12	1.37	3.73	0.96
12	1.13	1.57	4.41	0.96
13	0.96	0.34	1.69	0.94
14	1.15	0.79	3.18	0.96
15	1.01	0.84	2.55	0.95
16	1.18	1.69	6.51	0.97
17	1.12	1.34	3.68	0.96
18	1.20	1.70	7.41	0.97
19	1.19	1.39	5.93	0.97
20	1.12	0.71	2.82	0.96
21	0.27	0.04	0.51	0.10

range (0.55, 1.40) for the multivariate t, (0.86, 25.50) for the Laplace and (0.76, 0.99) for the contaminated normal. The Laplace distribution tends to heavily weight the observations closest to the mean. The contaminated normal distribution tends to assign very small weights to outliers and weights for nonoutliers are all close to 1.

Figure 3 also shows the mean and concentration ellipse obtained using ordinary least squares regression. It can be seen that the least squares regression results are significantly influenced by the extreme observations.

Table 2
Standard errors for the estimated coefficients

	x_1	x_2	x_3	x_4
t_4	8.53	0.11	0.29	0.11
$t_{1.1}$	4.28	0.06	0.15	0.06
Laplace	5.97	0.08	0.21	0.08
Contam.	8.43	0.11	0.29	0.11
$t_{1.1}$ (asymptotic)	4.7	0.054	0.147	0.063

NOTE: Asymptotic standard errors from Table 7 of Lange et al. (1989).

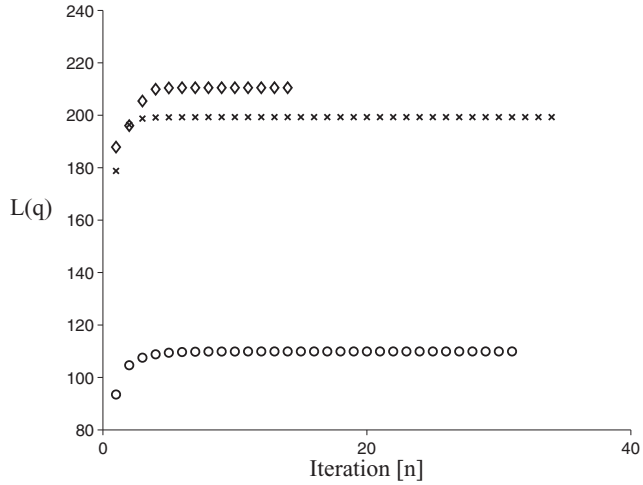


Figure 2. Computed values of the lower bound during the VB algorithm for t distribution (circle), Laplace distribution (cross) and contaminated normal distribution (diamond).

To gain some idea about the quality of the variational approximation, the computed means and 95% confidence intervals are compared to the results obtained using Gibbs sampler MCMC method. The Gibbs sampler is used to generate 5000 samples from the posterior distribution for each of the three families of robust regression distributions. The generated samples are used to obtain estimates for the mean and 95% confidence interval.

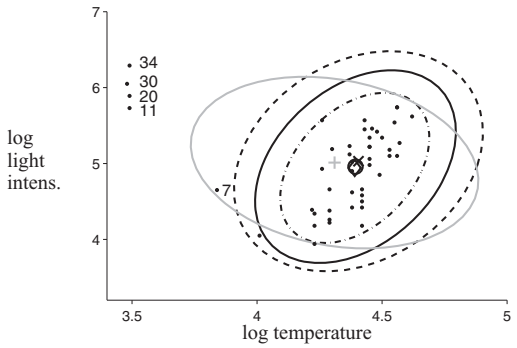


Figure 3. Fitted means and concentration ellipses for Student t distribution (circle, solid), Laplace distribution (cross, dashed), contaminated normal distribution (diamond, dash-dot) and least squares regression (gray plus, gray solid).

Table 3
Expected weights w_n for the observations

Observation	t_5	Laplace	Contam.
7	0.37	0.69	0.17
11	0.12	0.35	0.10
20	0.12	0.34	0.10
30	0.11	0.33	0.10
34	0.10	0.32	0.10

4.3. *Risk Research Data, Multivariate Observations, and Missing Data*

In the third example, we analyze St. Louis Risk Research data from Little and Rubin (2002, p. 119). The dataset contains information from 69 families with two children. The families are classified into three groups according to the mental health history of the parents. Group 1 is a control group of families, group 2 is a moderate risk group, and group 3 is a high-risk group. The data consist of standardized reading R and verbal V comprehension scores for both of the children in each family. For some children, the R or V or both are missing.

We wish to study if the reading and verbal scores in group 1 are better than in the combined group of 2 and 3. Each observation consists of four scores $\mathbf{Y}_n = [R_1, V_1, R_2, V_2]^T$, which are the reading and verbal comprehension scores for the first and second child, respectively. The missing values are assigned a value NaN. We form a regression model with eight regression parameters

$$\mathbf{Y}_n = \left[\, \mathbf{H}_1 \; \mathbf{H}_2 \,\right] \mathbf{x} + \mathbf{v},$$

where $\mathbf{H}_k = \mathbf{I}$, if the observation n was from group k and $\mathbf{H}_k = 0$ otherwise. The regression coefficients are

$$\mathbf{x} = \left[\, \mathbf{x}_1^T \; \mathbf{x}_2^T \,\right]^T = \left[\, R_1^{(1)} \; V_1^{(1)} \; R_2^{(1)} \; V_2^{(1)} \; R_1^{(2)} \; V_1^{(2)} \; R_2^{(2)} \; V_2^{(2)} \,\right]^T.$$

For the measurements, we consider multivariate t with four degrees of freedom, multivariate Laplace, and contaminated normal distribution with $\epsilon = 0.1$ and $c = 10$. We use the commands:

Table 4
Estimated means and 95% confidence intervals for $\mathbf{x}$

		x_1	95% CI	x_2	95% CI
t ($\nu = 5$)	VB	4.3937	(4.3478, 4.4338)	4.9591	(4.8062, 5.0782)
	MCMC	4.3934	(4.3414, 4.4414)	4.9604	(4.7969, 5.1182)
Laplace	VB	4.4056	(4.3718, 4.4395)	5.0296	(4.9309, 5.1283)
	MCMC	4.4067	(4.3582, 4.4546)	5.0362	(4.8617, 5.2054)
Cont. norm.	VB	4.3908	(4.3469, 4.4347)	4.9422	(4.7964, 5.0880)
	MCMC	4.3898	(4.3400, 4.4374)	4.9422	(4.7876, 5.0953)

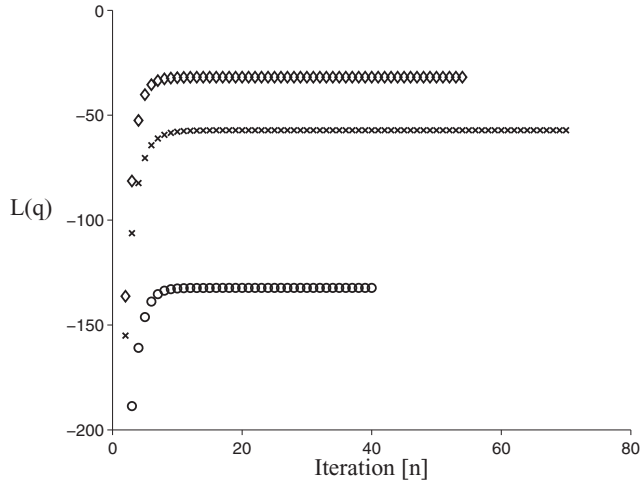


Figure 4. Computed values of the lower bound during the VB algorithm for t distribution ($\nu = 4$) (circle), Laplace distribution (cross), and contaminated normal distribution (diamond).

```
>> [m,Q,W,P,R,V]=rmvregress(H,Y);
>> [m,Q,W,P,R,V]=rmvregress(H,Y, 'laplace');
>> [m,Q,W,P,R,V]=rmvregress(H,Y, 'contnorm', [0.1, 10]);
```

The values of the lower bound during the VB updates are plotted in Fig. 4. We are interested in the difference in the means of the two groups, i.e., in the posterior distribution of $\mathbf{x}_1 - \mathbf{x}_2$. The joint posterior for $\mathbf{x} = [\begin{smallmatrix}\mathbf{x}1\\ \mathbf{x}_2\end{smallmatrix}]$is normal with mean $\mathbf{m}$ and variance $\mathbf{P}$. From this, it follows that $\mathbf{x}_1 - \mathbf{x}_2$ has a normal distribution with mean $\mathbf{m}_1 - \mathbf{m}_2$ and variance $\mathbf{P}_1 + \mathbf{P}_2 - \mathbf{P}_{1,2} - \mathbf{P}_{1,2}^T$. The marginal posteriors for each of the four differences are plotted in Fig. 5. The 95% posterior probability intervals are also listed in Table 5. The conclusions using all the distributions are that the scores are better in group 1. We see that our results are consistent with the results in Little and Rubin (2002, p. 261), where the posterior histograms

Table 5
95% posterior probability intervals for the difference $\mathbf{x}_1 - \mathbf{x}_2$. MCMC results from Little and Rubin (2002, p. 261)

	t_4 (MCMC)	t_4	Laplace	Contam.
$V_1^{(1)} - V_1^{(2)}$	(15.38, 43.99)	(18.04, 40.21)	(19.56, 38.98)	(13.58, 38.55)
$V_2^{(1)} - V_2^{(2)}$	(7.08, 31.42)	(8.25, 30.53)	(6.73, 26.44)	(5.47, 31.44)
$R_1^{(1)} - R_1^{(2)}$	(1.68, 17.94)	(3.75, 16.03)	(2.93, 13.72)	(3.69, 17.11)
$R_2^{(1)} - R_2^{(2)}$	(−0.60, 14.04)	(0.19, 12.26)	(0.62, 11.28)	(−0.92, 13.02)

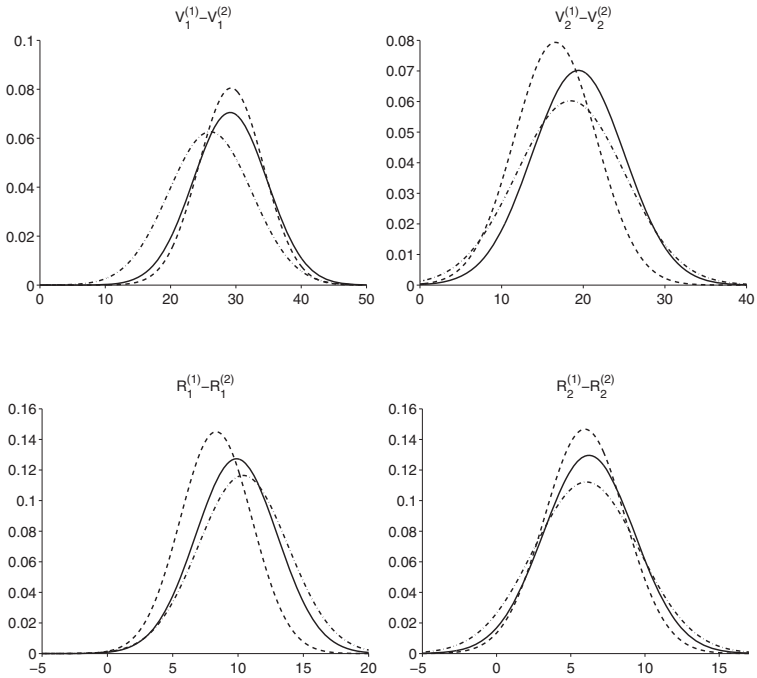


Figure 5. Marginal posterior distributions for the difference $\mathbf{x}_1 - \mathbf{x}_2$. Student t-distribution (solid), Laplace distribution (dashed), and contaminated normal distribution (dash-dot).

and 95% probability intervals were obtained using Monte Carlo methods for multivariate t distribution. As can be seen from the Monte Carlo results using the multivariate t distribution, the variational approximation for the posterior tends to underestimate the true posterior variance.

5. Conclusion

In this article, we considered multivariate robust linear regression with missing data. We concentrated on robust models in which the observations can be written as a Gaussian scale mixture. Specifically, we considered three families of distributions corresponding to different priors for the weights in the Gaussian scale mixture presentation. These families include as special cases the multivariate t, multivariate Laplace, and multivariate contaminated normal models that are often used for robust statistical modeling.

Robust linear regression is much studied in the statistical literature. Algorithms for statistical inference are mainly based on the EM algorithm or on Monte Carlo methods. In this article, we presented a variational Bayes method to compute an approximation for the posterior distribution. The VB method can be seen as an alternative to the computationally

heavy Monte Carlo methods in Bayesian inference. The VB algorithm is computationally comparable to the EM algorithm, but is better suited for full Bayesian inference because we obtain directly the full approximate posterior for the model parameters. The VB method can be also extended to include the estimation of the hyperparameters for the prior distribution of weights in the Gaussian scale mixture presentation.

Examples from literature were presented to illustrate the use of our Matlab/Octave implementation and to compare the results to those obtained in the references. The implementation can be freely downloaded from the Matlab Central file exchange.

The proposed method for variational inference is based on fully factorizing the posterior distribution, which also provides directly the posterior marginal distributions for all the variables. The accuracy of the approximation is in general difficult to access, however the variance of the approximated distribution tends to underestimate the true variance of the posterior (MacKay, 2003). This result is also observed for examples 2 and 3, where the confidence intervals obtained using MCMC methods tend to be wider than confidence intervals obtained using the variational method.

In this article, the lower bound was used mainly for inspecting the convergence of the VB algorithm. However, this can be also utilized for model comparison as discussed, for example, in (Beal, 2003, pp. 60–61). Also, differences of the VB method with the EM algorithm could be studied by comparing values of the lower bound with the values of log-likelihood during the iterations.

Interesting extension for the robust methods considered in this paper would be to apply them for nonlinear problems and time-series analysis. The variational method for nonlinear forward models using Gaussian noise is considered, for example, in Chappell et al. (2009). Combining these methods to deal with nonlinear models with the Gaussian scale mixture presentation of the measurement distribution could be used to obtain variational methods for the nonlinear robust regression. The variational inference has also recently been applied for robustifying the Kalman filter algorithm using *t*-distributed noise for the measurements (Agamennoni et al., 2011; Piché et al., 2012). Kalman filters with more general measurement distributions could be obtained by utilizing the Gaussian scale mixture models considered in this article.

Funding

The first author received financial support from the Tampere Doctoral Programme in Information Science and Engineering (TISE).

References

Abramowitz, M., Stegun, I. (1965). *Handbook of Mathematical Functions*. Mineola, New York: Dover Publications.

Agamennoni, G., Nieto, J. I., Nebot, E. M. (2011). An outlier-robust Kalman filter. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE. pp. 1551–1558.

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*. New York: Wiley.

Andrews, D. F., Mallows, C. L. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)* 36(1): 99–102.

Beal, M. J. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, London: University of London.

Beal, M. J., Ghahramani, Z. (2003). The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures. In *Bayesian Statistics*, Vol. 7. Oxford: Oxford University Press, pp. 453–463 .

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Berlin: Springer.

Bishop, C. M., Winn, J., Spiegelhalter, D. (2002). Vibes: A variational inference engine for Bayesian networks. In: Becker, S., Thrun, S., Obermayer, K., eds. *Advances in Neural Information Processing Systems XV*. Cambridge, Massachusetts: MIT Press.

Blattberg, R. C., Gonedes, N. J. (1974). A comparison of the stable and student distributions as statistical models for stock prices. *The Journal of Business* 47(2): 244–280.

Bloomfield, P., Steiger, W. L. (1983). *Least Absolute Deviations: Theory, Applications, and Algorithms*. Progress in probability and statistics. Boston: Birkhäuser.

Chappell, M., Groves, A., Whitcher, B., Woolrich, M. (2009). Variational Bayesian inference for a nonlinear forward model. *Signal Processing, IEEE Transactions on* 57(1): 223–236.

Dempster, A. P., Laird, N. M., Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1): 1–38.

Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B. (2003). *Bayesian Data Analysis*, 2nd ed. London: Chapman & Hall/CRC.

Hoeting, J., Hoeting, J., Raftery, A. E., Madigan, D. (1996). A method for simultaneous variable selection and outlier identification in linear regression. *Computational Statistics and Data Analysis* 22: 25–70.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics* 35(1): 73–101.

Jørgensen, B. (1982). *Statistical Properties of Generalized Inverse Gaussian distribution*. Lecture Notes in Statistics. Berlin: Springer.

Kotz, S., Kozubowski, T. J., Podgórski, K. (2001). *The Laplace Distribution and Generalizations*. Birkhäuser.

Kotz, S., Nadarajah, S. (2004). *Multivariate t Distributions and Their Applications*. Cambridge: Cambridge University Press.

Lange, K. L., Little, R. J. A., Taylor, J. M. G. (1989). Robust statistical modeling using the *t* distribution. *Journal of the American Statistical Association* 84(408): 881–896.

Little, R. J. A. (1988). Robust estimation of the mean and covariance matrix from data with missing values. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 37(1): 23–38.

Little, R. J. A., Rubin, D. B. (2002). *Statistical Analysis with Missing Data*. New York: Wiley.

Liu, C. (1996). Bayesian robust multivariate linear regression with incomplete data. *Journal of the American Statistical Association* 91(435): 1219–1227.

Liu, C., Rubin, D. B. (1994). The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. *Biometrika* 81(4): 633–648.

Liu, C., Rubin, D. B., Wu, Y. N. (1998). Parameter expansion to accelerate EM: The PX-EM algorithm. *Biometrika* 85(4): 755–770.

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge: Cambridge University Press.

McLachlan, G. J., Krishnan, T. (2008). *The EM Algorithm and Extensions*. 2nd ed. New York: Wiley.

Meng, X.-L., Dyk, D. v. (1997). The EM algorithm–an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society. Series B (Methodological)* 59(3): 511–567.

Meng, X.-L., Rubin, D. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* 80(2): 267–278.

Meng, X.-L., Rubin, D. B. (1991). Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm. *Journal of the American Statistical Association* 86(416): 899–909.

Ormerod, J., Wand, M. (2010). Explaining variational approximations. *The American Statistician* 64(2): 140–153.

Penny, W., Kilner, J., Blankenburg, F. (2007). Robust Bayesian linear models. *Neuroimage* 36(3): 661–671.

Phillips, R. (2002). Least absolute deviations estimation via the EM algorithm. *Statistics and Computing* 12: 281–285.

Piché, R., Särkkä, S., Hartikainen, J. (2012). Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate student-*t* distribution. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, IEEE. pp. 1–6.

Roberts, S., Penny, W. (2002). Variational Bayes for generalized autoregressive models. *Signal Processing, IEEE Transactions on* 50(9): 2245–2257.

Rousseeuw, P. J., Leroy, A. M. (2003). *Robust Regression and Outlier Detection*. New York: Wiley.

Rubin, D. B. (2004). Iteratively reweighted least squares. In *Encyclopedia os Statistical Sciences*. New York: Wiley.

Tanner, M. A., Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association* 82(398): 528–540.

Tipping, M., Lawrence, N. (2003). A variational approach to robust Bayesian interpolation. In *Neural Networks for Signal Processing, 2003. NNSP'03. 2003 IEEE 13th Workshop on*, pp. 229–238.

Titterington, D. (2004). Bayesian methods for neural networks and related models. *Statistical Science* 19(1): 128–139.

Tukey, J. W. (1960). A survey of sampling from contaminated distributions. In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, Stanford, CA: Stanford University Press. pp. 448–485.

Verboven, S., Hubert, M. (2005). LIBRA: a Matlab library for robust analysis. *Chemometrics and Intelligent Laboratory Systems* 75(2): 127–136.

Verdinelli, I., Wasserman, L. (1991). Bayesian analysis of outlier problems using the Gibbs sampler. *Statistics and Computing* 1: 105–117.

Wand, M. P., Ormerod, J. T., Padoan, S. A., Frührwirth, R. (2011). Mean field variational Bayes for elaborate distributions. *Bayesian Analysis* 6(4): 1–48.

West, M. (1984). Outlier models and prior distributions in Bayesian linear regression. *Journal of the Royal Statistical Society. Series B (Methodological)* 46(3): 431–439.

Zellner, A. (1976). Bayesian and non-Bayesian analysis of the regression model with multivariate student-t error terms. *Journal of the American Statistical Association* 71(354): 400–405.

Publication 3

Juha Ala-Luhtala, Simo Särkkä and Robert Piché. Gaussian filtering and variational approximations for Bayesian smoothing in continuous-discrete stochastic dynamic systems, *Signal Processing*, 111:(0):124-136, 2015

Signal Processing 111 (2015) 124–136

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Gaussian filtering and variational approximations for Bayesian smoothing in continuous-discrete stochastic dynamic systems

Juha Ala-Luhtala [a,*], Simo Särkkä [b], Robert Piché [c]

[a] Tampere University of Technology, Department of Mathematics, Korkeakoulunkatu 1, 33720 Tampere, Finland
[b] Aalto University, Department of Biomedical Engineering and Computational Science (BECS), Finland
[c] Tampere University of Technology, Department of Automation Science and Engineering, Finland

ARTICLE INFO

Article history:
Received 22 July 2014
Received in revised form
5 November 2014
Accepted 13 December 2014
Available online 20 December 2014

Keywords:
Bayesian smoothing
Gaussian approximation
Variational inference

ABSTRACT

The Bayesian smoothing equations are generally intractable for systems described by nonlinear stochastic differential equations and discrete-time measurements. Gaussian approximations are a computationally efficient way to approximate the true smoothing distribution. In this work, we present a comparison between two Gaussian approximation methods. The Gaussian filtering based Gaussian smoother uses a Gaussian approximation for the filtering distribution to form an approximation for the smoothing distribution. The variational Gaussian smoother is based on minimizing the Kullback–Leibler divergence of the approximate smoothing distribution with respect to the true distribution. The results suggest that for highly nonlinear systems, the variational Gaussian smoother can be used to iteratively improve the Gaussian filtering based smoothing solution. We also present linearization and sigma-point methods to approximate the intractable Gaussian expectations in the variational Gaussian smoothing equations. In addition, we extend the variational Gaussian smoother for certain class of systems with singular diffusion matrix.

1. Introduction

The continuous-discrete system refers to a system whose process dynamics are governed by a continuous-time stochastic differential equation (SDE) and whose measurements are taken at discrete time instants. These kinds of systems arise in many real-life applications. Some examples are navigation and tracking [1,2], weather forecasting [3], and systems biology [4]. Bayesian filtering and smoothing equations give the solution to the problem of estimating the state of the system from the noisy measurements. Computing the filtering and smoothing distributions involves solving the related partial differential equations [5,6], and is only tractable for linear-Gaussian systems (and some other special cases [7]). In this paper, we consider two different approaches for computing a Gaussian approximation for the smoothing distribution.

The solution of the optimal smoothing problem is closely connected to the optimal filtering problem. For linear Gaussian system the solution to the optimal filtering problem is given by the Kalman filter [8] in the discrete-time case and Kalman–Bucy filter [9] in the continuous-time case. The continuous-discrete Kalman filter is a combination of the two filters where the continuous-time prediction is used between observations and discrete-time update at observation times [5]. The continuous-time Rauch–Tung–Striebel smoother [10] uses the filtering solution and gives a recursive solution for the optimal smoothing problem in the continuous-discrete system. Other equivalent solutions to the optimal smoothing problem include, for example, the maximum likelihood smoother by Bryson and Frazier [11] and the two filter smoother by Fraser and Potter [12].

The classical Gaussian approximations for the nonlinear problem were based on linearizing the nonlinearities in

* Corresponding author. Tel.: +358 456509464.
E-mail address: juha.ala-luhtala@tut.fi (J. Ala-Luhtala).

http://dx.doi.org/10.1016/j.sigpro.2014.12.013
0165-1684/

the system using Taylor series based methods [11,5]. The Taylor series based methods can be seen as special cases of a more general Gaussian filtering and smoothing framework, where different filters and smoothers arise based on the numerical method for computing the Gaussian expectations [13,14]. A different approach for Gaussian smoothing was considered in [15–17], where a Gaussian approximation is sought by approximating the stochastic process giving the smoothed distribution with a linear process. The method is based on the fixed-form variational Bayes approximation [18] and minimizes the Kullback–Leibler divergence of the approximate distribution with respect to the true distribution.

The variational Gaussian approximation is considered further in [19–22]. Shen et al. [19] compared the variational approximation to a Monte Carlo Markov Chain (MCMC) solution for the one-dimensional double-well system and found that the variational method performed comparatively to the MCMC solution when the uncertainties in the measurements were not so large as to cause the true posterior to be multimodal. In [20] the variational Gaussian smoothing solution is used as a proposal distribution in an MCMC method to improve the efficiency of the algorithm. The variational MCMC method was found to outperform the hybrid Monte Carlo method for sparsely observed diffusion processes. Vrettas et al. [21,22] considered approximating the variational Gaussian smoothing equations using a radial basis function representation for the variational parameters. By imposing a certain structure for the variational parameter functions, the overall number of parameters to be optimized can be reduced.

The variational Gaussian smoothing equations derived by Archambeau et al. [15–17] require a non-singular effective diffusion matrix. Our first goal is to extend the variational Gaussian approximation to a certain class of singular models by considering an alternative derivation based on Girsanov's theorem.

Another problem in the variational Gaussian algorithm is the need to compute Gaussian expectations over nonlinear functions. Previous works on the variational approximation [15,16,19,21,22] have not presented details for computing the Gaussian expectations for general nonlinear systems. Our second goal is to extend the variational Gaussian smoothing method for general nonlinear systems by considering numerical approximations for the Gaussian expectations. The treatment is similar to the one in [14], where Taylor series based linearization, cubature, and unscented transform based sigma-point methods and Gauss–Hermite quadrature were used to compute Gaussian expectations in the Gaussian smoothers based on the classical Gaussian filtering framework.

In addition, we provide a comparison between the variational Gaussian approximation [15,16] and the Gaussian smoothers presented in [14]. Using a suitable change of variables the Gaussian smoothing equations can be converted to a variational form similar to the variational Gaussian smoothing equations. The computation of the Gaussian filtering based smoothing solution is numerically stable and provides good initial conditions for the variables in the variational Gaussian smoothing algorithm. This might help us to overcome the problems reported by Vrettas et al. [21] in the initialization of the variational Gaussian algorithm for high dimensional systems. Also, we study if the variational Gaussian smoother can be used to iteratively improve the results from the Gaussian filtering based smoother.

The organization of this paper is as follows. Section 2 presents the previous results for variational Gaussian smoother and the Gaussian smoother based on the Gaussian filtering framework for the continuous-discrete system. In Section 3, we extend the variational Gaussian smoothing equations for a certain class of singular models. In Section 4 we compare theoretically the Gaussian smoothers by presenting a conversion of the Gaussian filtering based smoothing equations to the variational form by a suitable change of variables. This also provides the initial values for the variational Gaussian smoothing algorithm. The problem of numerically computing the Gaussian expectations in the variational Gaussian smoothing equations is treated in Section 5. We present Taylor series linearization and sigma-point methods to approximate the Gaussian expectations in the variational Gaussian smoothing equations. The paper concludes with two synthetic-data examples that are used to compare the Gaussian smoothers, and also provides comparison of the different numerical methods for the computation of the Gaussian expectations.

1.1. Problem statement

The continuous-discrete system considered in this paper is given by

$$dx = f(x,t)\,dt + L(t)\,d\beta(t), \quad t \in \mathbb{R}^+ \tag{1}$$

$$y_k = h_k(x(t_k)) + v_k, \tag{2}$$

where $x(t) \in \mathbb{R}^n$ is the state, $f(x(t),t)$ is the drift term, and $\beta(t)$ is a Brownian motion stochastic process with diffusion matrix $Q(t)$. The effective diffusion matrix for the process is given by

$$\Sigma(t) = L(t)Q(t)L^T(t). \tag{3}$$

The initial conditions are assumed to be normally distributed $x(t_0) \sim N(m_0, P_0)$. The measurement noise $\{v_k\}$ is a zero mean Gaussian white noise sequence with covariance matrix R_k. The measurement noise $\{v_k\}$, process noise $\beta(t)$ and initial conditions $x(t_0)$ are assumed to be mutually independent.

Let $y_1, \ldots, y_K$ be the measurements taken at discrete time instants $t_1, \ldots, t_K$. The solution to the Bayesian smoothing problem is the posterior distribution:

$$p(x(t)|y_1, \ldots, y_K), \quad t \in [t_0, t_K]. \tag{4}$$

In this paper, we concentrate on Gaussian approximations for the smoothing distribution. That is, the smoothing distribution is approximated as

$$p(x(t)|y_1, \ldots, y_K) \approx \mathrm{N}(x(t)|m(t), P(t)), \tag{5}$$

where $m(t)$ is the mean function and $P(t)$ is the autocovariance $P(t,t')$ at $t = t'$ for the approximating distribution. The smoothing problem now reduces to finding the expressions for the mean and covariance functions.

Table 1
Summary of variational Gaussian smoother (VGS) and Gaussian filtering based Gaussian smoother (GFGS) algorithms.

VGS	GFGS
1. Initialize $A^{(0)}(t)$ and $b^{(0)}(t)$ and set $k=0$.	1. Compute $m_f(t)$ and $P_f(t)$ by solving ordinary differential equations (20) and (21) from t_0 to t_K. At observation times, do measurement update using Eqs. (22)–(25).
2. Compute $m^{(k)}(t)$ and $P^{(k)}(t)$ by solving ordinary differential equations (7) and (8) from t_0 to t_K.	2. Compute $m_s(t)$ and $P_s(t)$ by solving ordinary differential equations (26) and (27) from t_K to t_0.
3. Compute $\lambda^{(k)}(t)$ and $\Psi^{(k)}(t)$ by solving ordinary differential equations (9) and (10) from t_K to t_0. At observation times use jump conditions (14) and (15).	
4. Update $A^{(k+1)}(t)$ and $b^{(k+1)}(t)$ using Eqs. (17) and (18).	
5. If not converged yet, set $k=k+1$ and go to Step 2.	

2. Gaussian smoothing for continuous-discrete systems

This section presents the background for the variational Gaussian smoother and Gaussian filtering based Gaussian smoother. The two algorithms are summarized in Table 1 and the details are presented in the following subsections.

2.1. Variational Gaussian approximation

The variational Gaussian approximation for the continuous-discrete smoothing problem was derived by Archambeau et al. [15–17]. The method is based on approximating the smoothing process with a linear process:

$$dx = [-A(t)+b(t)]\,dt + \sqrt{\Sigma(t)}\,d\beta(t), \tag{6}$$

where $A(t)$ and $b(t)$ are parameters of the approximation and $\beta(t)$ is a Brownian stochastic process with the identity diffusion matrix. The solution to the linear SDE (6) is a Gaussian process. The marginal density at each time is given by $q(x(t)) = \mathrm{N}(x(t)|m(t), P(t))$, where the mean and the covariance are computed from the ordinary differential equations:

$$\frac{d}{dt}m(t) = -A(t)m(t)+b(t), \tag{7}$$

$$\frac{d}{dt}P(t) = -A(t)P(t)-P(t)A^T(t)+\Sigma(t). \tag{8}$$

The SDE in Eq. (6) defines a Gaussian measure over the sample paths that approximates the true posterior measure. To fit the parameters $A(t)$ and $b(t)$ a cost function between probability measures is needed. The cost function used by Archambeau et al. [15,16] is the Kullback–Leibler (KL) divergence between the approximating probability measure with respect to the true probability measure. Minimizing this form of KL-divergence also minimizes the variational free energy [16] and maximizes the lower bound to the log-marginal probability of the data [18, p. 479].

The optimal parameters in terms of the KL-divergence are then shown to satisfy equations [7,8]

$$\frac{d}{dt}\lambda(t) = A^T(t)\lambda(t) - \nabla_m \mathbb{E}_q[e(x,t)], \tag{9}$$

$$\frac{d}{dt}\Psi(t) = \Psi(t)A(t)+A^T(t)\Psi(t) - \nabla_P \mathbb{E}_q[e(x,t)], \tag{10}$$

$$A(t) = -\mathbb{E}_q[\nabla_x f(x,t)]^T + 2\Sigma(t)\Psi(t), \tag{11}$$

$$b(t) = \mathbb{E}_q[f(x,t)] + A(t)m(t) - \Sigma(t)\lambda(t), \tag{12}$$

where $\lambda(t)$ and $\Psi(t)$ are auxiliary Lagrange functions and

$$e(x,t) = \frac{1}{2}[f(x,t)+A(t)x-b(t)]^T\Sigma^{-1}(t)\big[f(x,t)+A(t)x-b(t)\big]. \tag{13}$$

Note that for notational convenience we have suppressed the time dependance of x and use $x = x(t)$. At observation times, the Lagrange functions satisfy jump conditions:

$$\lambda(t_k^+) = \lambda(t_k^-) + \nabla_m \mathbb{E}_q[u_k(x)], \tag{14}$$

$$\Psi(t_k^+) = \Psi(t_k^-) + \nabla_P \mathbb{E}_q[u_k(x)], \tag{15}$$

where

$$u_k(x) = \frac{1}{2}[y_k - h_k(x)]^T R_k^{-1}\big[y_k - h_k(x)\big]. \tag{16}$$

To find a solution satisfying the Euler–Lagrange equations, we propose here a slight modification of the iterative algorithm given by Archambeau et al. [15,16]. Given the previous estimates $A^{(k)}(t)$ and $b^{(k)}(t)$, the mean and covariance differential equations (7) and (8) are solved forward in time from t_0 which gives $m^{(k+1)}(t)$ and $P^{(k+1)}(t)$. Using these, the Lagrange differential equations (9) and (10) are then solved backward in time from t_K which gives $\lambda^{(k+1)}(t)$ and $\Psi^{(k+1)}(t)$. New estimates $A^{(k+1)}(t)$ and $b^{(k+1)}(t)$ are then computed by using a damped fixed-point update:

$$A^{(k+1)}(t) = A^{(k)}(t) + \gamma_k(A(t) - A^{(k)}(t)), \tag{17}$$

$$b^{(k+1)}(t) = b^{(k)}(t) + \gamma_k(b(t) - b^{(k)}(t)), \tag{18}$$

where $A(t)$ and $b(t)$ are computed using Eqs. (11) and (12) respectively, and $\gamma_k \in (0,1)$ is a damping parameter that is used to prevent numerical instabilities caused by too large updates. Instead of using constant damping parameter as in [15], we propose to select the parameter γ_k so that the objective function and therefore also the KL-divergence is reduced at each time step. The parameter γ_k can then be chosen by decreasing the value until reduction in KL-divergence is achieved or by using a more sophisticated approximate line search method. In a computer implementation of the variational Gaussian smoother, the values of the functions $A(t)$ and $b(t)$ are computed and stored at discrete time points. The differential equations can be

solved using any standard numerical solver such as Euler or Runge–Kutta methods.

2.2. Gaussian filtering based Gaussian smoother

The Gaussian filtering based Gaussian smoother uses the Gaussian approximation for the filtering distribution to form the Gaussian approximation for the smoothing distribution [14].

The Gaussian filtering (or Gaussian assumed density filtering) approach is well known in the literature (see e.g. [23,5,24]) and uses the approximation:

$$p(x(t)|y_1,\ldots,y_k) \approx \mathrm{N}(x(t)|m_f(t),P_f(t)), \tag{19}$$

where $t\in[t_k,t_{k+1})$ and $p(x(t)|y_1,\ldots,y_k)$ is the filtering distribution. The mean and covariance functions are recursively computed using the following prediction and update steps. In the prediction step the mean and covariance functions are propagated from time t_{k-1} to time t_k using

$$\frac{dm_f}{dt} = \mathbb{E}_f\left[f(x,t)\right], \tag{20}$$

$$\frac{dP_f}{dt} = \mathbb{E}_f\left[(x-m_f)f(x,t)^T\right] + \mathbb{E}_f\left[f(x,t)(x-m_f)^T\right] + \Sigma(t). \tag{21}$$

In the update step, the information from the latest measurement y_k is used to update the predicted estimates $m(t_k^-)$ and $P(t_k^-)$ using equations

$$S_k = \mathbb{E}_f[(h_k(x)-\mathbb{E}_f[h_k(x)])(h_k(x)-\mathbb{E}_f[h_k(x)])^T] + R_k, \tag{22}$$

$$K_k = \mathbb{E}_f[(x-m_f(t_k^-))(h_k(x)-\mathbb{E}_f[h_k(x)])^T]S_k^{-1}, \tag{23}$$

$$m_f(t_k) = m_f(t_k^-) + K_k(y_k-\mathbb{E}_f[h_k(x)]), \tag{24}$$

$$P_f(t_k) = P_f(t_k^-) - K_kS_kK_k^T. \tag{25}$$

Särkkä and Sarmavuori [14] extend the general Gaussian filtering ideas also to smoothing problems and derive three types of Gaussian smoothers labeled as Type I, Type II and Type III. The Type I smoother is derived by using the Gaussian approximation for the filtering distribution to approximate the exact partial differential equations for the smoothed mean and covariance. The Type II smoother is derived by discretizing the dynamic model, applying the discrete-time smoothing equations and then taking the limit as the discretization time approaches zero. Formulating the Type II smoothing equations into a computationally efficient form gives the Type III smoother.

The Type I smoothing equations are numerically quite sensitive, which causes the Type I smoother to diverge quite often compared to the Type II and Type III smoothers. Also, the Type I smoother is computationally more demanding and was not found to be clearly better than the Type II and Type III smoothers in the synthetic-data example considered in [14]. For this reason we decided to concentrate on the Type II and Type III smoothers in this paper.

The Type II smoothing equations are given by

$$\frac{dm_s}{dt} = \mathbb{E}_f\left[f(x,t)\right] + \left[\mathbb{E}_f\left[f(x,t)(x-m_f)^T\right] + \Sigma(t)\right]P_f^{-1}(m_s-m_f), \tag{26}$$

$$\frac{dP_s}{dt} = \left(\mathbb{E}_f\left[f(x,t)(x-m_f)^T\right] + \Sigma\right)P_f^{-1}P_s + P_sP_f^{-1}(\mathbb{E}_f[f(x,t)(x-m)^T]^T + \Sigma(t)) - \Sigma(t). \tag{27}$$

If the Jacobian $F_x(x,t)$ of $f(x,t)$ is available, we can alternatively use

$$\mathbb{E}_f[f(x,t)(x-m_f)^T] = \mathbb{E}_f[F_x(x,t)]P_f \tag{28}$$

in the filtering and smoothing equations. The expectations in the smoothing equations are with respect to the filtering density, which means that they can be computed already during the filtering stage. This is used in the Type III smoothing equations, which reformulate the filtering and Type II smoothing equations so that no expectations need to be computed during the smoothing stage (see [14] for details).

3. Variational Gaussian smoothing for singular models

To compute the term $e(x,t)$ in Eq. (13), a nonsingular effective diffusion matrix $\Sigma(t)$ is required. In this section we extend the variational Gaussian smoothing for certain class of nonsingular models. If $\Sigma(t)$ is singular, we can partition the state vector to subvectors x_1 and x_2, so that the dynamics of x_1 are described by an ordinary differential equation and x_2 by a SDE with nonsingular diffusion matrix. We concentrate here on the case that the differential equation for x_1 is linear and our model can therefore be formulated as

$$\frac{dx_1}{dt} = F_1(t)x, \tag{29}$$

$$dx_2 = f_2(x,t)\,dt + \sqrt{\Sigma_2(t)}\,d\beta, \tag{30}$$

where $\Sigma_2(t)$ is a nonsingular diffusion matrix and the augmented state is $x^T=[x_1^T\ x_2^T]$. As an example, where these kinds of models arise, consider a moving object with random disturbances in its equation of motion. The dynamics can be modeled with an SDE $dv = f(r,v,t)\,dt + \sqrt{\Sigma(t)}\,d\beta$, where r and v denote the objects position and velocity respectively. Forming the state space model for the state $x^T=[r^T,\ v^T]$ then results in a model of the form of in Eqs. (29) and (30).

Denote by n_1 and n_2 the dimensions of the state vectors x_1 and x_2 respectively. We now seek an approximate smoothing process of the form

$$\frac{dx_1}{dt} = F_1(t)x, \tag{31}$$

$$dx_2 = (-A(t)x + b(t))\,dt + \sqrt{\Sigma_2(t)}\,d\beta, \tag{32}$$

where $A(t)$ and $b(t)$ are the variational parameters. This gives a Gaussian process, where mean $m(t)$ and covariance $P(t)$ follow the differential equations

$$\frac{d}{dt}m(t) = -\tilde{A}(t)m(t) + \tilde{b}(t), \tag{33}$$

$$\frac{d}{dt}P(t) = -\tilde{A}(t)P(t) - P(t)\tilde{A}^T(t) + \Sigma(t), \tag{34}$$

with

$$\tilde{A}(t)=\begin{bmatrix}-F_1(t)\\ A(t)\end{bmatrix},\quad \tilde{b}(t)=\begin{bmatrix}0_{n_1\times 1}\\ b(t)\end{bmatrix},\quad \Sigma(t)=\begin{bmatrix}0_{n_1\times n_1} & 0_{n_1\times n_2}\\ 0_{n_2\times n_1} & \Sigma_2(t)\end{bmatrix}. \tag{35}$$

The KL-divergence term for the singular system can be computed using Girsanov's theorem and the optimal solution minimizing the KL-divergence then satisfies (see [25] and Appendix A for details)

$$\frac{d}{dt}\lambda(t)=\tilde{A}^T(t)\lambda(t)-\nabla_m\mathbb{E}_q[e(x,t)], \tag{36}$$

$$\frac{d}{dt}\Psi(t)=\Psi(t)\tilde{A}(t)+\tilde{A}^T(t)\Psi(t)-\nabla_P\mathbb{E}_q[e(x,t)], \tag{37}$$

$$A(t)=-\mathbb{E}_q[F_{2,x}(x,t)]+2\Sigma_2(t)M\Psi(t), \tag{38}$$

$$b(t)=\mathbb{E}_q[f_2(x,t)]+A(t)m(t)-\Sigma_2(t)M\lambda(t), \tag{39}$$

where $F_{2,x}(x,t)$ is the Jacobian of $f_2(x,t)$ and

$$e(x,t)=\frac{1}{2}[f_2(x,t)+A(t)x(t)-b(t)]^T\Sigma_2^{-1}(t)[f_2(x,t)+A(t)x(t)-b(t)]. \tag{40}$$

The matrix M selects the relevant part of the Lagrange parameter functions and is given by

$$M=\begin{bmatrix}0_{n_1\times n_2} & I_{n_2\times n_2}\end{bmatrix}. \tag{41}$$

At observation times, the Lagrange multipliers satisfy the boundary conditions given by Eqs. (14) and (15).

4. Differences between the Gaussian smoothers

In this section, we study the differences between the variational and Gaussian filtering based Gaussian smoothers. First we introduce a change of variables that converts the Type II Gaussian smoothing equations to a form similar to the variational Gaussian smoothing equations. This conversion is similar to the results from Rauch et al. [10], where it was shown that the Rauch–Tung–Striebel smoothing equations are formally equivalent to the smoothing equations presented by Bryson and Frazier [11].

The conversion is achieved by using the change of variables:

$$\lambda(t)=-P_f^{-1}(t)(m_s(t)-m_f(t)),\quad \Psi(t)=-\frac{1}{2}\left(P_f^{-1}(t)-P_s^{-1}(t)\right). \tag{42}$$

Inserting $\lambda(t)$ and $\Psi(t)$ to Eqs. (26) and (27) and computing the time derivatives of $\lambda(t)$ and $\Psi(t)$ give (see Appendix B)

$$\frac{d}{dt}m_s(t)=-A(t)m_s(t)+b(t), \tag{43}$$

$$\frac{d}{dt}P_s(t)=-A(t)P(t)-P(t)A^T(t)+\Sigma(t), \tag{44}$$

$$\frac{d}{dt}\lambda(t)=A^T(t)\lambda(t)-2\Psi(t)\Sigma(t)\lambda(t), \tag{45}$$

$$\frac{d}{dt}\Psi(t)=\Psi(t)A(t)+A^T(t)\Psi(t)-2\Psi(t)\Sigma(t)\Psi(t), \tag{46}$$

where

$$A(t)=-\mathbb{E}_f[F_x(x,t)]+2\Sigma(t)\Psi(t), \tag{47}$$

$$b(t)=\mathbb{E}_f[f(x,t)]+\mathbb{E}_f[F_x(x,t)](m_s(t)-m_f(t))+A(t)m_s(t)-\Sigma(t)\Psi(t). \tag{48}$$

For linear measurement function $h_k(x)=H_kx$, the measurement update for $\lambda(t)$ and $\Psi(t)$ in the variational form of the Type II smoother is the same as in the variational Gaussian smoother and is given by

$$\lambda(t_k^+)=\lambda(t_k^-)+H_k^TR_k^{-1}(m_s(t_k)-y_k) \tag{49}$$

$$\Psi(t_k^+)=\Psi(t_k^-)+\frac{1}{2}H_k^TR_k^{-1}H_k. \tag{50}$$

For nonlinear measurement function, the measurement updates for $\lambda(t)$ and $\Psi(t)$ are in general not equal to the variational Gaussian smoother update in Eqs. (14) and (15).

Similarities to the variational Gaussian smoothing equations are evident from Eqs. (43)–(48). Note that Eqs. (47) and (48) for the parameters $A(t)$ and $b(t)$ are otherwise similar to the variational Gaussian smoothing equations (11) and (12), but the function $f(x,t)$ is replaced with statistical linearization with respect to the filtering distribution:

$$f(x,t)\approx\mathbb{E}_f[f(x,t)]+\mathbb{E}_f[F_x(x,t)](x-m_f(t)). \tag{51}$$

Furthermore, using the statistical linearization (51) to approximate the gradients in Eqs. (9) and (10) gives

$$\nabla_m\mathbb{E}[e(x,t)]\approx 2\Psi(t)\Sigma(t)\lambda(t), \tag{52}$$

$$\nabla_P\mathbb{E}[e(x,t)]\approx 2\Psi(t)\Sigma(t)\Psi(t). \tag{53}$$

That is, the differential equations (45) and (46) can be seen as an approximation to the exact differential equations (9) and (10) in the variational Gaussian smoother. This suggests that for linear measurements, the Type II Gaussian smoother can be seen to approximate the variational Gaussian smoother by using statistical linearization with respect to the filtering distribution.

We can use the Type II Gaussian smoothing solution as an initial iterand for the variational parameters by computing $A^{(0)}(t)$ and $b^{(0)}(t)$ using Eqs. (47) and (48), where $\lambda(t)$ and $\Psi(t)$ are given by (42). In [21] it was noted that the iterative solution of the variational Gaussian smoothing equations is sensitive to the initial values of the variational parameters $A(t)$ and $b(t)$. This way, the variational Gaussian smoother can be seen as an iterative way to improve the Type II Gaussian smoothing solution. The benefit of using the variational Gaussian smoother to improve the Type II smoother results is studied further in the synthetic-data examples.

5. Computation of Gaussian expectations in the variational Gaussian smoother

The Gaussian smoothers considered in this paper require computations of Gaussian expectations over arbitrary nonlinear functions. For some simple models these can be computed analytically, but for many models no analytical expressions exist. The computation of Gaussian expectations for the Gaussian filtering based smoothers is considered in

[14]. In this work we concentrate on computing the expectations in the variational Gaussian smoothing equations and present the extended, cubature, unscented, and Gauss–Hermite forms of the variational Gaussian smoother.

Gaussian expectations need to be computed in the differential equations (9) and (10) for the functions $\lambda(t)$ and $\Psi(t)$, in Eqs. (11) and (12) for the variational parameter functions $A(t)$ and $b(t)$, and in the measurement update in Eqs. (14) and (15). In order to avoid computing derivatives of the drift function $f(x,t)$ and of the measurement function $h_k(x)$, the gradients with respect to $m(t)$ and $P(t)$ can be written in the form (see Appendix C)

$$\nabla_m \mathbb{E}[e(x,t)] = P^{-1}\mathbb{E}[e(x,t)(x-m)], \tag{54}$$

$$\nabla_P \mathbb{E}[e(x,t)] = \frac{1}{2}P^{-1}\mathbb{E}\left[e(x,t)(x-m)(x-m)^T\right]P^{-1} - \frac{1}{2}\mathbb{E}[e(x,t)]P^{-1}. \tag{55}$$

For the measurement updates, the gradients are computed similarly with $e(x,t)$ replaced by $u_k(x)$.

If the Jacobians $F_x(x,t)$ and $H_{k,x}(x)$ of $f(x,t)$ and $h_k(x)$, respectively, are available, the gradients can be alternatively written in the form

$$\nabla_m \mathbb{E}[e(x,t)] = \mathbb{E}[e_x(x,t)], \tag{56}$$

$$\nabla_P \mathbb{E}[e(x,t)] = \frac{1}{2}P^{-1}\mathbb{E}[e_x(x,t)(x-m)^T]^T, \tag{57}$$

where

$$e_x(x,t) = \nabla_x e(x,t) = [F_x(x,t) + A(t)]^T \Sigma^{-1}(t)[f(x,t) + A(t)x(t) - b(t)]. \tag{58}$$

To compute the gradients in the measurement update, $e_x(x,t)$ is replaced with

$$u_{k,x}(x) = \nabla_x u_k(x) = H_{k,x}^T(x) R_k^{-1}[h_k(x) - y_k]. \tag{59}$$

5.1. Taylor series based linearization

In the extended Kalman filter and smoother, the Gaussian expectations are computed by using a first order Taylor series linearization. Proceeding similarly, we use the following approximations for the extended variational Gaussian smoother:

$$f(x,t) \approx f(m,t) + F_x(m,t)(x(t) - m(t)), \tag{60}$$

$$h_k(x) \approx h_k(m) + H_{k,x}(m)(x(t)) - m(t)), \tag{61}$$

where $F_x(m,t)$ and $H_{k,x}(m)$ are the Jacobians of $f(x,t)$ and $h_k(x)$, respectively, evaluated at the current mean estimate $m(t)$. Using this approximation, the expectations in Eqs. (11) and (12) are given by

$$\mathbb{E}[f(x,t)] \approx f(m,t), \tag{62}$$

$$\mathbb{E}[F_x(x,t)] \approx F_x(m,t). \tag{63}$$

The expectations needed in the gradients are given by

$$\nabla_m \mathbb{E}[e(x,t)] \approx (F_x(m,t) + A(t))^T \Sigma^{-1}(t)(f(m,t) + A(t)m(t) - b(t)), \tag{64}$$

$$\nabla_P \mathbb{E}[e(x,t)] \approx \frac{1}{2}(F_x(m,t) + A(t))^T \Sigma^{-1}(t)(F_x(m,t) + A(t)), \tag{65}$$

and for the measurement update

$$\nabla_m \mathbb{E}[u_k(x)] \approx H_{k,x}^T(m) R_k^{-1}(t)(h_k(m) - y_k), \tag{66}$$

$$\nabla_P \mathbb{E}[u_k(x)] \approx \frac{1}{2}H_{k,x}^T(m)^T R_k^{-1}(t) H_{k,x}(m). \tag{67}$$

5.2. Sigma-point methods

General sigma-point rule computes the Gaussian expectations using the approximation

$$\mathbb{E}[g(x,t)] \approx \sum_i W^{(i)} g(m + \sqrt{P}\xi_i, t), \tag{68}$$

where the weights $W^{(i)}$ and vectors ξ_i are chosen depending on the used sigma-point method. In this paper we consider the cubature, unscented and Gauss–Hermite sigma-point methods. The number of sigma-points in cubature and unscented transform methods are $2n$ and $2n+1$ respectively. The Gauss–Hermite integration uses s^n sigma-points, where s is a parameter that gives the order of the used Hermite polynomial. Details for computing the vectors ξ_i and weights $W^{(i)}$ are given in [14] for the cubature and unscented methods and in [26,24,13] for the Gauss-Hermite method.

The sigma-point approximation for expectations in Eqs. (11) and (12) is given by

$$\mathbb{E}[f(x,t)] \approx \sum_i W^{(i)} f(m + \sqrt{P}\xi_i, t), \tag{69}$$

$$\mathbb{E}[F_x(x,t)] = \mathbb{E}[f(x,t)(x-m)^T]P^{-1} \approx \sum_i W^{(i)} f(m + \sqrt{P}\xi_i, t)\xi_i^T \sqrt{P}^{-1}. \tag{70}$$

The general sigma-point approximations for the gradients are given by

$$\nabla_m \mathbb{E}[e(x,t)] \approx \sum W^{(i)} e(m + \sqrt{P}\xi_i, t)\sqrt{P}^{-1}\xi_i, \tag{71}$$

$$\nabla_P \mathbb{E}[e(x,t)] \approx \frac{1}{2}\sum W^{(i)} e\left(m + \sqrt{P}\xi_i, t\right)\sqrt{P}^{-T}\left(\xi_i\xi_i^T - I\right)\sqrt{P}^{-1}. \tag{72}$$

The expectations needed in the observation updates (14) and (15) are computed with $e(x,t)$ replaced by $u_k(x)$.

The sigma-point approximation for the alternative forms (56) and (57) of the gradients is given by

$$\nabla_m \mathbb{E}[e(x,t)] \approx \sum W^{(i)} e_x(m + \sqrt{P}\xi_i), \tag{73}$$

$$\nabla_P \mathbb{E}[e(x,t)] \approx \frac{1}{2}\sum W^{(i)} \sqrt{P}^{-T}\xi_i e_x^T\left(m + \sqrt{P}\xi_i, t\right). \tag{74}$$

The measurement updates are computed similarly with $e_x(x,t)$ replaced by $u_{k,x}(x)$. The cubature, unscented, and Gauss–Hermite forms of the variational Gaussian smoother are then obtained by using the corresponding choice for the weights $W^{(i)}$ and vectors ξ_i (cf. [14,26]).

For a linear drift function $f(x,t)$, the term $\mathbb{E}[e(x,t)(x-m)(x-m)^T]$ is a fourth order polynomial. For this reason, the sigma-point rules that are only accurate up to a third order monomial (cubature and unscented rule) give

generally a poor approximation of this expectation. Therefore, for cubature and unscented sigma-point methods, the use of the alternative form given by Eqs. (73) and (74) is recommended.

6. Numerical experiments

The Gaussian smoothers are compared using two different synthetic-data experiments. The tests are done by first running the Gaussian filtering based Gaussian smoother (GFGS) and then using the result as the initial condition for the variational Gaussian smoother (VGS). The VGS iteration is terminated when the absolute change in the KL-divergence between successive iterations is less than 10^{-3}.

In the first experiment, a one-dimensional double-well system is used. The same system was also used to demonstrate the VGS in [15,19,21]. A 5-dimensional reentry problem is used for the second experiment. This system was used to test the continuous-discrete unscented Kalman filter in [27] and demonstrates the use of VGS for singular systems. Also, for this system the computation of the needed Gaussian expectations is not possible analytically.

The metrics used to compare the estimates given by the Gaussian smoothers are the root mean square error (RMSE), negative log-likelihood (NLL) and 95% consistency. The RMSE and NLL are given by equations

$$\mathrm{RMSE} = \sqrt{\frac{1}{t_K - t_0}\int_{t_0}^{t_K} \|x(t) - m(t)\|^2 \, dt} \tag{75}$$

$$\mathrm{NLL} = \frac{1}{t_K - t_0}\int_{t_0}^{t_K} \ln N\big(x(t)\big|m(t), P(t)\big)\, dt, \tag{76}$$

where $x(t)$ is the true state, $m(t)$ is the estimated mean, and $P(t)$ is the estimated covariance. The 95% consistency is defined as the fraction of times the true state is inside the 95% ellipsoid of $N(m(t), P(t))$. The values of the continuous time metrics are computed using the values of $m(t)$ and $P(t)$ computed at discrete time points.

The different approximation methods for computing the Gaussian expectations in the smoothing equations are also compared. The tested methods are labeled as

- *EXT*: The method using the Taylor series based linearization.
- *CT*: Cubature rule based sigma-point method.
- *UT*: Unscented rule based sigma-point method with parameter values $\alpha=1$, $\beta=2$, and $\kappa=0$.
- *G–H*: Gauss–Hermite series based sigma-point method with order 3.

The methods labeled CT2, UT2, and G-H2 use the respective sigma-point methods with the alternative formulation given in Eqs. (73) and (74).

The algorithms are implemented in Matlab version R2013b and run on a Macbook Pro with 2.8 GHz Intel Core i7.

6.1. Double-well

The double-well system is given by

$$dx = 4x(1-x^2)\, dt + \sqrt{\sigma}\, d\beta, \quad y_k = x(t_k) + v_k, \tag{77}$$

where the measurement noise v_k is the white zero-mean Gaussian with variance R. The prior distribution for the double-well system is non-Gaussian and multimodal, but the smoothing distribution can be reasonably well approximated with a Gaussian provided that the measurement variance is not too large. The modes are located at $x=1$ and $x=-1$ and for sufficiently large value of the process noise parameter σ, there is frequent transition from one mode to the other.

The Gaussian smoothers are compared for 4 different values for the measurement variance R. For each value of the measurement variance R, a data set of 100 Monte Carlo simulations is generated using Euler–Maruyama discretization with time step $\Delta t = 0.01$ from $t_0 = 0$ to $t=10$. The process noise parameter is chosen to be $\sigma = 1$, which is sufficiently large to cause frequent transition between the modes. The initial state is chosen as $x(0) \sim N(0, 1)$.

For this system, the Gaussian expectations needed in the Gaussian smoothers can be computed analytically (see e.g. [15]). For both Gaussian smoothers, the differential equations are solved using 4th order Runge–Kutta method with time step 0.01. Average number of 35 iterations was observed for the VGS when using the GFGS results as initial conditions.

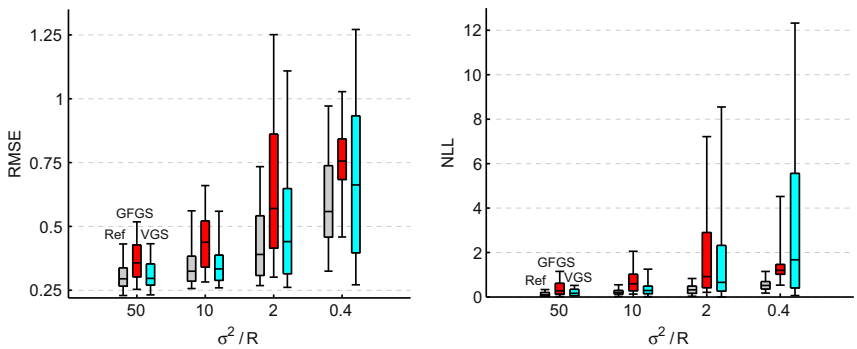


Fig. 1. The RMSE (left) and NLL (right) for the reference (grey), GFGS (red) and VGS (cyan) for different values of the measurement variance R. The Gaussian expectations in the smoothing equations are computed analytically. The boxplots show the 5%, 25%, 50%, 75% and 95% quantiles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

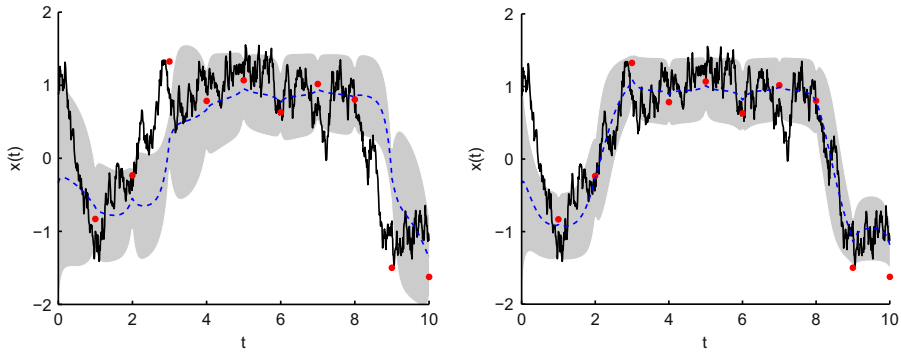


Fig. 2. The estimated mean (dashed, blue) and 95%-confidence region (shaded) for GFGS (left) and VGS (right). True state (solid, black) and measurements (red dot) are also shown. The Gaussian expectations in the smoothing equations are computed analytically. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 2
The mean 95% consistencies for the GFGS and VGS for different values of measurement noise R. The Gaussian expectations in the smoothing equations are computed analytically.

Method	Ratio σ^2/R			
	50	10	2	0.4
GFGS	0.93	0.89	0.81	0.91
VGS	0.91	0.88	0.80	0.74

Also, we observed a much faster convergence using this initialization than with the naive initialization using just the initial conditions.

The boxplots of the RMSE and NLL results for different values of the measurement variance R are shown in Fig. 1. For comparison, a reference smoothing solution is also computed using a finite difference approximation of the exact Bayesian smoothing equations [28]. For the reference solution, the NLL is computed from the finite difference approximation for the smoothing density.

For small values of the measurement variance, the VGS results are very close to the reference solution and clearly outperforms the GFGS approach in terms of RMSE and NLL. The relatively poor RMSE values for the GFGS are due to the poor estimation of the transitions between the two modes. A typical time series for measurement variance $R=0.1$ is shown in Fig. 2.

For measurement variance $R=2.5$, the true posterior is bimodal and the VGS tends to have the estimated mean close to one of the modes with relatively small variance. This results in very large NLL values for the VGS, when the true path is not close to the mode. In comparison, the GFGS tends to have the mean close to zero with the 95%-confidence region covering both modes. This shows us very good NLL values and smaller spread of the RMSE values compared to the VGS.

The mean 95%-consistency results over the 100 Monte Carlo simulations are shown in Table 2. From the consistency results, we see that in general the VGS tends to underestimate the variance compared to the GFGS. This is especially clear in the $R=2.5$ case. The underestimation of the variance is a general property of the variational type of approximations [29, p. 431] .

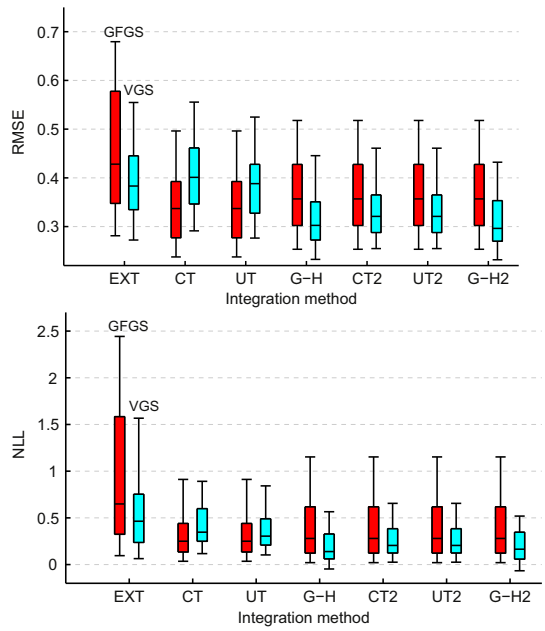


Fig. 3. The RMSE (top) and NLL (bottom) for the GFGS (red) and VGS (cyan) using different methods to compute the Gaussian expectations. The boxplots show the 5%, 25%, 50%, 75% and 95% quantiles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

The different approximation methods for the Gaussian expectations were compared for measurement variance $R=0.02$ and using the same data set of 100 Monte Carlo simulations as in the first experiment. The VGS failed to converge to a solution on 5 cases using the EXT method, but no failures were observed using the other methods.

The boxplots of the RMSE and NLL values for the GFGS and VGS when using the different methods to approximate the Gaussian expectations are shown in Fig. 3. The VGS using the EXT and G–H methods clearly improves the

results of the corresponding GFGS in terms of RMSE and NLL. Numerical problems were observed for the VGS using CT and UT rules, which resulted in poor RMSE and NLL values compared to the GFGS results. Using the alternative formulation of CT2 and UT2 works clearly better and slightly improve the results of the GFGS using the CT and UT methods. The VGS using G–H2 method gives results nearly identical to the VGS using the exact Gaussian expectations.

The mean 95%-consistency values over the 100 Monte Carlo simulations are shown in Table 3. The VGS using the EXT method shows a slight improvement in the consistency compared to the corresponding GFGS result. For the sigma-point methods, the differences are smaller with GFGS giving in general slightly better consistency results.

The mean computational times for the different methods are given in Table 4. The computational time of the VGS method depends on the number of iterations. The computation time of 1 VGS iteration varies from 1 to 2 times the computation time of the GFGS for the EXT, CT2, UT2, G–H and G–H2 methods. The increased time in the VGS iteration is mainly caused by the computation time for the linesearch step.

For comparison, we also included CT2, UT2 and G–H2 versions for the GFGS that use Eq. (28) to compute the Gaussian expectations. There seems to be no significant improvement in using the alternative formulation for the GFGS.

6.2. Reentry

The state $x=[r,v,\alpha]^T$ of the reentry problem consists of the vehicle's position r and velocity v in a 2-dimensional coordinate system and a parameter α of its aerodynamic properties. The dynamics are given by

$$d\begin{bmatrix} r(t) \\ v(t) \\ \alpha(t) \end{bmatrix} = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} & 0_{1\times 1} \\ G(x,t)I_{2\times 2} & D(x,t)I_{2\times 2} & 0_{1\times 1} \\ 0_{1\times 2} & 0_{1\times 2} & 0_{1\times 1} \end{bmatrix} \begin{bmatrix} r(t) \\ v(t) \\ \alpha(t) \end{bmatrix} dt + \begin{bmatrix} 0_{2\times 3} \\ I_{3\times 3} \end{bmatrix} d\beta(t), \tag{78}$$

where the gravity related force term $G(x,t)$ and drag related force term $D(x,t)$ are given by

$$G(x,t) = -\frac{Gm_0}{\|r(t)\|^3}, \tag{79}$$

$$D(x,t) = -\beta_0 e^{\alpha} \exp\left\{\frac{R_0 - \|r(t)\|}{H_0}\right\} \|v(t)\|. \tag{80}$$

The diffusion matrix for the Brownian motion $\beta(t)$ is

$$Q(t) = \begin{bmatrix} 2.4064\times 10^{-5} & 0 & 0 \\ 0 & 2.4064\times 10^{-5} & 0 \\ 0 & 0 & 1\times 10^{-6} \end{bmatrix}. \tag{81}$$

The effective diffusion matrix for this model is singular and the dynamic model can be written in the form of Eqs. (29) and (30)). The values $\beta_0 = -0.59783$, $H_0 = 13.406$, $Gm_0 = 3.9860\times 10^5$ and $R_0 = 6374$ are used as typical values for the parameters [27] (see [30]).

A radar located at $s=[s_x,s_y]^T$ periodically measures the range and bearing of the vehicle with 1 Hz frequency. The measurement model is given by

$$y_k = \begin{bmatrix} \|r(t_k) - s\| \\ \tan^{-1}\left(\frac{r_2(t_k)-s_y}{r_1(t_k)-s_x}\right) \end{bmatrix} + v_k, \tag{82}$$

where the measurement noise v_k is the white zero-mean Gaussian with covariance matrix:

$$R_k = \begin{bmatrix} 1\times 10^{-3} & 0 \\ 0 & 1.7\times 10^{-3} \end{bmatrix}. \tag{83}$$

The state trajectory and noisy measurements are simulated from $t_0 = 0$ to $t_K = 200$ using Euler–Maruyama discretization with time-step $\Delta t = 0.01$. The initial state is drawn from a Gaussian prior with mean and covariance given by

$$m(t_0) = \begin{bmatrix} 6500.4 & 349.14 & -1.8093 & -6.7967 & 0.6932 \end{bmatrix}^T, \tag{84}$$

$$P(t_0) = \begin{bmatrix} 10^{-6}\cdot I_{4\times 4} & 0_{4\times 1} \\ 0_{1\times 4} & 0 \end{bmatrix}. \tag{85}$$

For this model the computation of the Gaussian expectations needed in the GFGS and VGS is not possible analytically. The Gaussian expectations were computed using the EXT, CT, UT, and G–H integration rules. For the VGS it was necessary to use the alternative formulation of CT2 and UT2 rules, since the CT and UT rules caused numerical problems and failure of the algorithm to converge. For G–H this was not a problem. The differential equations were solved using the standard 4th order Runge–Kutta method with an integration step of

Table 3
The mean 95% consistencies for the GFGS and VGS using different methods to compute the Gaussian expectations.

Method	EXT	CT	UT	G–H	CT2	UT2	G–H2
GFGS	0.83	0.94	0.94	0.93	0.93	0.93	0.93
VGS	0.86	0.93	0.94	0.93	0.94	0.94	0.91

Table 4
The mean computational times in seconds for Gaussian filtering based and variational Gaussian smoothers. The number in parenthesis shows the mean number of iterations for the VGS method.

Method	EKF	CT	UT	G–H	CT2	UT2	G–H2
GFGS	0.3	0.5	0.6	1.9	0.5	0.6	1.9
VGS	4.6 (10)	3.3 (2)	3.2 (2)	17.3 (9)	11.9 (14)	12.3 (14)	44.3 (21)

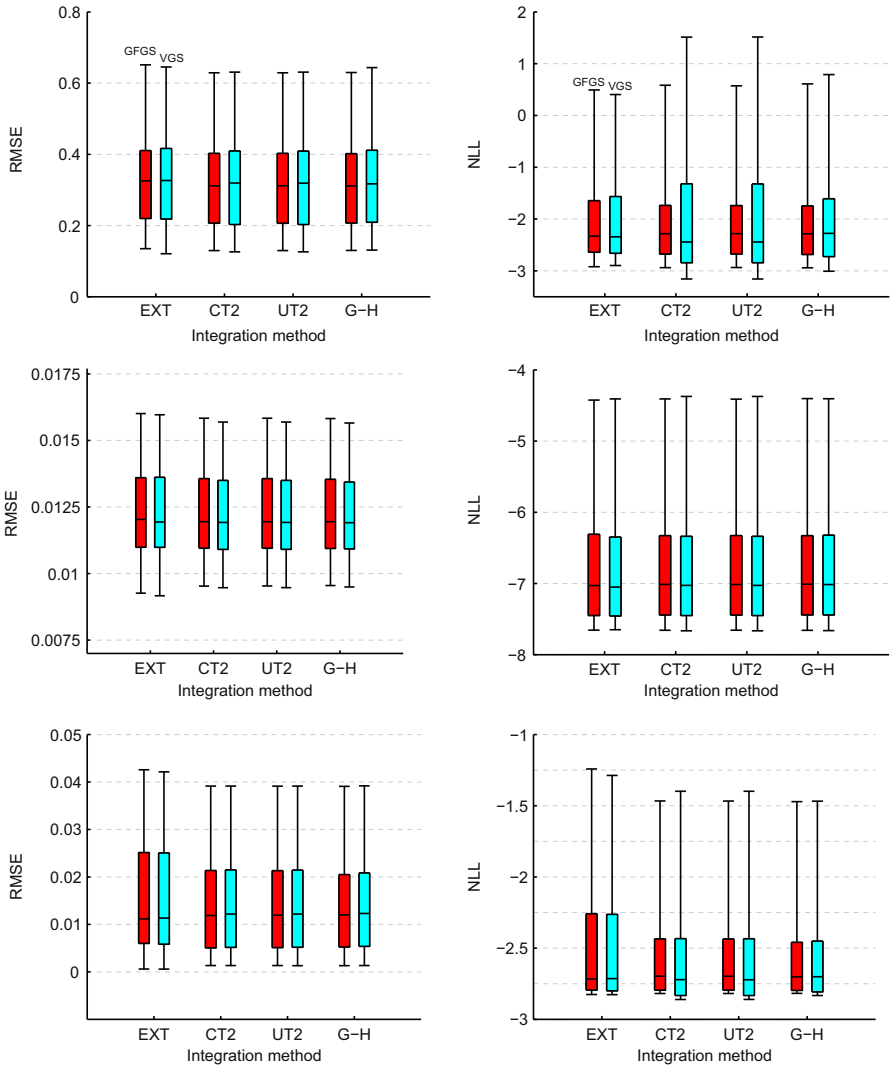


Fig. 4. The RMSE and NLL values for the Gaussian filtering based (red) and variational (cyan) Gaussian smoothers using different methods to compute the Gaussian expectations. The top row shows the results for position, the middle row for velocity and the bottom row for the aerodynamic parameter. The boxplots show the 5%, 25%, 50%, 75% and 95% quantiles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

0.1. The initial mean and covariance for the smoothers are given by Eq. (85), except that we used $m_5(t_0) = 0$ and $P_{5,5}(t_0) = 1$ for the unknown aerodynamic parameter. Note also that when simulating the true trajectory, the process noise $Q_{3,3}$ for the aerodynamic parameter is set to zero. For computational purposes, a pseudo-noise term is added when running the Gaussian smoothing algorithms (cf. [27,31]).

The boxplots of RMSE and NLL results for 100 Monte Carlo simulations are shown in Fig. 4. No clear difference in the RMSE results can be seen between GFGS and VGS methods. In the NLL results, some difference can be seen for the position, where the upper quantiles are better for the GFGS using sigma-point methods. Comparing the methods to compute Gaussian expectations, we see that the sigma-point methods give better results than the Taylor series linearization for the aerodynamic parameter, but no clear difference can be seen in the results for position and velocity. The VGS using sigma-point methods gives slightly more compact variance estimate for the position than the GFGS that can be seen in the mean 95%-consistency results in Table 5. Only a small decrease of KL-divergence was observed, which explains the small difference between the methods. The average computation times are shown in Table 6. As in the first example, the time for one iteration of VGS varies from 1 to 2 times the time for GFGS with the same integration method.

Table 5
The mean 95% consistencies for the Gaussian filtering based and variational Gaussian smoothers.

Method	Position	Velocity	Parameter
EXT-GSGS	0.94	0.95	0.95
EXT-VGS	0.94	0.95	0.95
CT-GSGS	0.95	0.95	0.97
CT2-VGS	0.89	0.95	0.96
UT-GSGS	0.95	0.95	0.97
UT2-VGS	0.89	0.95	0.96
GH-GSGS	0.95	0.95	0.97
GH-VGS	0.93	0.95	0.96

Table 6
The mean computational times in seconds for Gaussian filtering based and variational Gaussian smoothers. The number in the parenthesis shows the mean number of iterations for the VGS method.

Method	EKF	CT2	UT2	G-H
GFGS	1.8	2.0	2.3	14.7
VGS	7.7 (4)	12.9 (6)	14.0 (6)	59.3 (4)

7. Discussion and conclusions

The Gaussian filtering based Gaussian smoother provides good initial conditions for the variational Gaussian smoother and often solves the problem with initialization mentioned in [21]. The variational Gaussian smoother provides Gaussian approximation that is optimal in the sense that it minimizes the Kullback–Leibler divergence of the approximating distribution with respect to the true distribution. However, the examples considered here show that this will not always improve the estimate with respect to other commonly used metrics.

Using statistical linearization with respect to the filtering distribution in the variational Gaussian smoothing equations gives formally the Gaussian filtering based smoother as a special case. This suggests that for highly nonlinear systems, the variational Gaussian smoother could better capture the nonlinearities. This is demonstrated in the numerical experiment for the double-well system, where the variational Gaussian smoother clearly improves the Gaussian filtering based smoother estimate for small measurement variances. However, no clear improvement was observed in the second numerical experiment for the reentry problem. The reason could be that the nonlinearities in the reentry system are not high enough to gain benefit from using the variational Gaussian smoother equations. A drawback of the variational Gaussian smoother is that it tends to underestimate the variance compared to the Gaussian filtering based smoother.

For general nonlinear systems the Gaussian expectations can be computed using Taylor series based linearization or standard sigma-point methods. The Taylor series based variational Gaussian smoother linearizes the drift and measurement function with respect to the current mean estimate and is therefore similar in idea to the iterated extended Kalman smoother [32]. Using unscented transform and cubature rule based sigma-point methods in the variational Gaussian smoother resulted in some numerical problems. This is because the unscented and cubature methods are only accurate up to third order monomials, but even for linear systems some of the expectations are over fourth order polynomials. The numerical problems can be reduced by computing the Jacobians of the drift and measurement functions and using an alternative form for the expectations. Also, a higher order unscented transform could be used for these expectations (see [27,33]). Third order Gauss–Hermite integration rule worked well for both examples considered in this paper, but the computational cost is high especially for high-dimensional systems. For some high-dimensional systems the computational cost of the Gauss–Hermite method could be reduced by using Rao–Blackwellization [34].

Compared to the Gaussian filtering based Gaussian smoother the variational Gaussian smoother is more complex to implement and is computationally heavier. The computation time of variational Gaussian smoother depends on the number of iterations needed before convergence. For a real time implementation, the computational cost can be reduced, for example, by relaxing the convergence conditions or using a small fixed number of iterations. The obtained estimate would not necessarily be then optimal, but would still reduce the KL-divergence compared to the initial estimate.

Acknowledgments

The first author received financial support from the Tampere Doctoral Programme in Information Science and Engineering (TISE) and Tampere University of Technology Doctoral Programme in Engineering and Natural Sciences. The second author would like to thank the Academy of Finland (Projects 266940 and 273475).

Appendix A. Computing the KL-divergence using Girsanov's theorem

This section presents the derivation of the KL-divergence term for the system with singular effective diffusion matrix. The KL-divergence term can be partitioned to [17]

$$\mathrm{KL}(\mathbb{Q}_X \,\|\, \mathbb{P}_{X|Y}) = \mathrm{KL}(\mathbb{Q}_X \,\|\, \mathbb{P}_X) - \sum_{k=1}^{K} \mathbb{E}_q[\ln p(y_k|x_k)], \tag{A.1}$$

where $\mathbb{P}_X$ corresponds to the joint probability law of the stochastic processes $x_1(t)$ and $x_2(t)$, and $\mathbb{Q}_X$ to the joint probability law of the stochastic processes $s_1(t)$ and $s_2(t)$ defined by

$$\frac{dx_1}{dt} = F_1(t)x, \quad dx_2 = f_2(x(t),t)\,dt + d\beta(t)$$
$$\frac{ds_1}{dt} = F_1(t)s, \quad ds_2 = g_2(x(t),t)\,dt + d\beta(t),$$

where $\beta(t)$ is the Brownian motion with diffusion matrix $Q(t)$ with respect to measure $\mathbb{P}_X$.

The processes $s_1(t)$ and $s_2(t)$ are weak solutions to the original system under the measure $\mathbb{Q}_X$ that is defined through the Radon–Nikodym derivative [25]:

$$\mathbb{E}\left[\frac{d\mathbb{Q}_X}{d\mathbb{P}_X}\middle| F_t\right] = Z(t),$$

where F_t is the natural filtration of the Brownian motion $\beta(t)$ and

$$Z(t)=\exp\Bigg[\int_0^t \{f_2(s_1(t),s_2(t),t)-g_2(s_1(t),s_2(t),t)\}^T\, d\beta(t) -\frac{1}{2}\int_0^t \{f_2(s_1(t),s_2(t),t)-g_2(s_1(t),s_2(t),t)\}^T Q^{-1}(t) \times\{f_2(s_1(t),s_2(t),t)-g_2(s_1(t),s_2(t),t)^T\}\Bigg].$$

The KL-divergence term in the right side of Eq. (A.1) is then given by

$$\begin{aligned}\mathrm{KL}(\mathbb{Q}_X\,\|\,\mathbb{P}_X)&=\mathbb{E}_{\mathbb{Q}_X}\left[-\ln Z(t)\right]\\&=\frac{1}{2}\int_0^t \mathbb{E}_q\Big[\{f_2(s_1(t),s_2(t),t)-g_2(s_1(t),s_2(t),t)\}^T Q^{-1}(t)\\&\quad\times\{f_2(s_1(t),s_2(t),t)-g_2(s_1(t),s_2(t),t)\}\Big].\end{aligned}$$

Inserting $g_2(s_1(t),s_2(t),t)=-A(t)s(t)+b(t)$ gives then the KL-divergence:

$$\mathrm{KL}(\mathbb{Q}_X\,\|\,\mathbb{P}_{X|Y})=\int_{t_0}^{t_K}\mathbb{E}_q\left[e(x,t)+\sum_{k=1}^{K}u_k(x)\delta(t-t_k)\right]dt, \quad \text{(A.2)}$$

where $u_k(x)$ is given by Eq. (16) and

$$e(x,t)=\frac{1}{2}[f_2(x,t)+A(t)x(t)-b(t)]^T\Sigma_2^{-1}(t)\left[f_2(x,t)+A(t)x(t)-b(t)\right]. \quad \text{(A.3)}$$

The objective function is given by

$$\mathcal{F}(A,b,m,P)=\int_{t_0}^{t_K}\left\{\mathbb{E}_q\left[e(x,t)+\sum_{k=1}^{K}u_k(x)\delta(t-t_k)\right] -\lambda(t)^T\left(\frac{dm}{dt}+\tilde{A}(t)m(t)-\tilde{b}(t)\right) -\mathrm{trace}\left[\Psi(t)\left(\frac{dP}{dt}+\tilde{A}(t)P(t)+P(t)\tilde{A}(t)^T-\Sigma(t)\right)\right]\right\}dt. \quad \text{(A.4)}$$

Forming the Euler–Lagrange equations as in [15,16] for this model gives the desired solution.

Appendix B. Converting the Gaussian filtering based Gaussian smoother to the variational form

Here we present the derivation of the variational form of the Gaussian filtering based Gaussian smoother. This is achieved by using the change of variables:

$$\lambda=-P_f^{-1}(m_s-m_f),\quad \Psi=-\frac{1}{2}\left(P_f^{-1}-P_s^{-1}\right).$$

Computing the time derivatives of the new variables and inserting the filtering and smoothing differential equations (20) and (21), and (26) and(27) gives

$$\begin{aligned}\frac{d}{dt}\lambda&=P_f^{-1}\left(\frac{d}{dt}P_f\right)P_f^{-1}(m_s-m_f)-P_f^{-1}\left(\frac{d}{dt}m^s-\frac{d}{dt}m_f\right)\\&=P_f^{-1}(\mathbb{E}_f[F_x(x)]P_f+P_f\mathbb{E}[F_x^T(x)]+Q)P_f^{-1}(m_s-m_f)\\&\quad-P_f^{-1}(\mathbb{E}_f[f(x)]+\mathbb{E}_f[F_x(x)](m_s-m_f)+QP_f^{-1}(m_s-m_f)-\mathbb{E}_f[f(x)])\\&=-\mathbb{E}_f[F_x(x)]^T\lambda,\end{aligned}$$

and

$$\begin{aligned}\frac{d}{dt}\Psi&=\frac{1}{2}P_f^{-1}\left(\frac{d}{dt}P_f\right)P_f^{-1}-\frac{1}{2}P_s^{-1}\left(\frac{d}{dt}P_s\right)P_s^{-1}\\&=\frac{1}{2}P_f^{-1}(\mathbb{E}_f[F_x(x)]P_f+P_f\mathbb{E}[F_x(x)]^T+Q)P_f^{-1}\\&\quad-\frac{1}{2}P_s^{-1}\left(\mathbb{E}_f[F_x(x)]P_s+QP_f^{-1}P_s+P_s\mathbb{E}_f\left[F_x^T(x)\right]+P_sP_f^{-1}Q-Q\right)P_s^{-1}\\&=-\Psi\mathbb{E}_f[F_x(x)]-\mathbb{E}_f[F_x(x)]^T\Psi+2\Psi Q\Psi.\end{aligned}$$

Inserting $A(t)=-\mathbb{E}_f[F_x(x)]+2\Sigma(t)\Psi(t)$ to the above equations gives

$$\frac{d}{dt}\lambda(t)=A^T(t)\lambda(t)-2\Psi(t)\Sigma(t)\lambda(t),$$
$$\frac{d}{dt}\Psi(t)=\Psi(t)A(t)+A^T(t)\Psi(t)-2\Psi(t)\Sigma(t)\Psi(t).$$

Appendix C. Gradients with respect to mean and covariance

In this section we derive the expressions for the gradients with respect to the mean vector m and covariance matrix P of a Gaussian expectation over a scalar function $e(x)$. These are used to form the sigma-point approximations for the Gaussian expectations in the variational Gaussian smoothing equations. Computing the gradients gives

$$\begin{aligned}\nabla_m\mathbb{E}[e(x)]&=\nabla_m\left[\int e(x)N(x|m,P)\,dx\right]=\int e(x)\nabla_m N(x,|m,P)\,dx\\&=\int e(x)N(x,|m,P)P^{-1}(x-m)\,dx\\&=P^{-1}\mathbb{E}[e(x)(x-m)]=\mathbb{E}[\nabla_x e(x)],\end{aligned}$$

and

$$\begin{aligned}\nabla_P\mathbb{E}[e(x)]&=\nabla_P\left[\int e(x)N(x|m,P)\,dx\right]=\int e(x)\nabla_P N(x|m,P)\,dx\\&=\int e(x)N(x|m,P)\frac{1}{2}\left[P^{-1}(x-m)(x-m)^TP^{-1}-P^{-1}\right]dx\\&=\frac{1}{2}P^{-1}\mathbb{E}\left[e(x)(x-m)(x-m)^T\right]P^{-1}-\frac{1}{2}\mathbb{E}[e(x)]P^{-1}\\&=\frac{1}{2}P^{-1}\mathbb{E}\left[\nabla_x e(x)(x-m)^T\right].\end{aligned}$$

References

[1] M. Grewal, L. Weill, A. Andrews, Global Positioning Systems, Inertial Navigation, and Integration, Wiley, Hoboken, New Jersey, 2001.
[2] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software, John Wiley & Sons, Hoboken, New Jersey, 2004.
[3] E. Kalnay, Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, Cambridge, UK, 2003.
[4] D. Wilkinson, Stochastic Modelling for System Biology, CRC Press, Boca Raton, FL, 2006.
[5] A.H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.
[6] C.T. Leondes, J.B. Peller, E.B. Stear, Nonlinear smoothing theory, IEEE Trans. Syst. Sci. Cybern. 6 (1) (1970) 63–71.
[7] F.E. Daum, Exact finite-dimensional nonlinear filters, IEEE Trans. Autom. Control 31 (7) (1986) 616–622.
[8] R.E. Kalman, A new approach to linear filtering and prediction problems, J. Fluids Eng. 82 (1) (1960) 35–45.
[9] R.E. Kalman, R.S. Bucy, New results in linear filtering and prediction theory, J. Basic Eng. 83 (1) (1961) 95–108.
[10] H.E. Rauch, F. Tung, C.T. Striebel, Maximum likelihood estimates of linear dynamic systems, J. Am. Inst. Aeronaut. Astronaut. 3 (8) (1965) 1445–1450.

[11] A. Bryson, M. Frazier, Smoothing for linear and nonlinear dynamic systems, in: Proceedings of the Optimum System Synthesis Conference, 1963, pp. 353–364.
[12] D. Fraser, J. Potter, The optimum linear smoother as a combination of two optimum linear filters, IEEE Trans. Autom. Control 14 (4) (1969) 387–390, http://dx.doi.org/10.1109/TAC.1969.1099196.
[13] Y. Wu, D. Hu, M. Wu, X. Hu, A numerical-integration perspective on Gaussian filters, IEEE Trans. Signal Process. 54 (8) (2006) 2910–2921.
[14] S. Särkkä, J. Sarmavuori, Gaussian filtering and smoothing for continuous-discrete dynamic systems, Signal Process. 93 (2) (2013) 500–510.
[15] C. Archambeau, D. Cornford, M. Opper, J. Shawe-Taylor, Gaussian process approximations of stochastic differential equation, J. Mach. Learn. Res.: Workshop Conf. Proc. 11 (2007) 1–16.
[16] C. Archambeau, M. Opper, Y. Shen, D. Cornford, J.S. Shawe-taylor, Variational inference for diffusion processes, in: J. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems, vol. 20, Curran Associates, Inc., Red Hook, NY. 2008, pp. 17–24.
[17] C. Archambeau, M. Opper, Approximate Inference for Continuous-time Markov Processes, Bayesian Time Series Models, Cambridge University Press, Cambridge, UK. 2011, pp. 125–140.
[18] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[19] Y. Shen, C. Archambeau, D. Cornford, M. Opper, J. Shawe-Taylor, R. Barillec, A comparison of variational and Markov chain Monte Carlo methods for inference in partially observed stochastic dynamic systems, J. Signal Process. Syst. 61 (1) (2010) 51–59.
[20] Y. Shen, D. Cornford, M. Opper, C. Archambeau, Variational Markov chain Monte Carlo for Bayesian smoothing of non-linear diffusions, Comput. Stat. 27 (1) (2012) 149–176.
[21] M.D. Vrettas, D. Cornford, M. Opper, Y. Shen, A new variational radial basis function approximation for inference in multivariate diffusions, Neurocomputing 73 (7) (2010) 1186–1198.
[22] M.D. Vrettas, D. Cornford, M. Opper, Estimating parameters in stochastic systems: a variational Bayesian approach, Physica D: Nonlinear Phenom. 240 (23) (2011) 1877–1900.
[23] H. Kushner, Dynamical equations for optimal nonlinear filtering, J. Differ. Equ. 3 (2) (1967) 179–190.
[24] K. Ito, K. Xiong, Gaussian filters for nonlinear filtering problems, IEEE Trans. Autom. Control 45 (5) (2000) 910–927.
[25] S. Särkkä, T. Sottinen, Application of Girsanov theorem to particle filtering of discretely observed continuous-time non-linear systems, Bayesian Anal. 3 (3) (2008) 555–584.
[26] S. Särkkä, Bayesian Filtering and Smoothing, Cambridge University Press, Cambridge, UK, 2013.
[27] S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation, Proc. IEEE 92 (3) (2004) 401–422.
[28] S. Särkkä, Recursive Bayesian inference on stochastic differential equations (Doctoral dissertation), Helsinki University of Technology, 2006.
[29] D.J.C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, Cambridge, UK, 2003.
[30] J. Austin, C. Leondes, Statistically linearized estimation of reentry trajectories, IEEE Trans. Aerosp. Electron. Syst. 17 (1) (1981) 54–61.
[31] S.J. Julier, J.K. Uhlmanm, Correction to "Unscented Filtering and Nonlinear Estimation" ⟨http://www.gatsby.ucl.ac.uk/byron/nlds/julier04corr.pdf⟩, 2004.
[32] B.M. Bell, The iterated Kalman smoother as a Gauss–Newton method, SIAM J. Optim. 4 (3) (1994) 626–636.
[33] U.N. Lerner, Hybrid Bayesian networks for reasoning about complex systems (Ph.D. thesis), Stanford University, 2002.
[34] P. Closas, C. Fernández-Prades, The marginalized square-root Quadrature Kalman filter, in: IEEE Eleventh International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2010, pp. 1–5.

Publication 4

Juha Ala-Luhtala, Nick Whiteley, Kari Heine, Robert Piché. An introduction to twisted particle filters and parameter estimation in non-linear state-space models, *IEEE Transactions on Signal Processing*, 64:(18):4875-4890, 2016

An Introduction to Twisted Particle Filters and Parameter Estimation in Non-linear State-space Models

Juha Ala-Luhtala, Nick Whiteley, Kari Heine and Robert Piché, *Senior Member, IEEE,*

***Abstract*—Twisted particle filters are a class of sequential Monte Carlo methods recently introduced by Whiteley and Lee [1] to improve the efficiency of marginal likelihood estimation in state-space models. The purpose of this article is to extend the twisted particle filtering methodology, establish accessible theoretical results which convey its rationale, and provide a demonstration of its practical performance within particle Markov chain Monte Carlo for estimating static model parameters. We derive twisted particle filters that incorporate systematic or multinomial resampling and information from historical particle states, and a transparent proof which identifies the optimal algorithm for marginal likelihood estimation. We demonstrate how to approximate the optimal algorithm for non-linear state-space models with Gaussian noise and we apply such approximations to two examples: a range and bearing tracking problem and an indoor positioning problem with Bluetooth signal strength measurements. We demonstrate improvements over standard algorithms in terms of variance of marginal likelihood estimates and Markov chain autocorrelation for given CPU time, and improved tracking performance using estimated parameters.**

***Index Terms*—Particle filter, sequential Monte Carlo, particle MCMC, Gaussian state-space model, parameter estimation.**

I. Introduction

State-space models are applied to a wide variety of signal processing problems, especially in positioning, tracking and navigation [2]–[4]. These models need to be calibrated by inferring unknown parameters from data. There are a variety of approaches to this inference problem, such as maximum likelihood (ML) or maximum a posteriori (MAP) estimation using the Expectation Maximization algorithm or Laplace approximations, Gaussian filtering based approximations, and state augmentation techniques [3], [5], [6]. In this paper we consider a Bayesian approach, which has the advantage of allowing prior information about parameters to be imparted, and a variety of estimates and measures of uncertainty to be reported based on the posterior distribution. By using a Markov chain Monte Carlo (MCMC) algorithm, e.g. Metropolis-Hastings (M-H) (see [7] for an introduction), one can in principle explore the entire posterior, but in practice the design of efficient MCMC algorithm can be a challenging task.

A direct application of M-H to a state-space model requires evaluation of the marginal likelihood of data, which is a high-dimensional, analytically intractable integral in many cases of interest. However, this issue can be circumvented through the application of pseudo-marginal MCMC methods [8], [9], which allow MCMC algorithms yielding samples from the desired posterior to be constructed if an unbiased estimator of the marginal likelihood is available. Particle filters [10] (see [3], [4] for overviews in the context of tracking applications) provide such an estimator, and the resulting MCMC scheme is known as a particle Markov chain Monte Carlo (PMCMC) method [11]. Typically the most substantial contribution to the overall cost of a PMCMC algorithm arises from the need to run a particle filter at each iteration of the MCMC sampler, and the performance of the sampler is sensitive to the variability of the marginal likelihood estimate which the particle filter delivers [12]. This motivates the development of particle filters which can provide reliable marginal likelihood estimates at a low computational cost.

In this paper we develop new "twisted particle filtering" methodology, building from ideas recently introduced by [1]. Twisted particle filters are purposefully designed to provide more reliable approximations of marginal likelihoods than standard particle filters, while preserving the lack-of-bias property which permits their use within PMCMC.

Unlike traditional approaches to improving the efficiency of particle filters which modify the proposal distribution on a per-particle basis [13] or employ auxiliary weights for resampling [14], twisted particle filters are derived by applying a form of re-weighting to the particle system as a whole, using a so-called "twisting" function. The role of the twisting function is to incorporate information from the observations, possibly future and past, into the mechanism by which particles are propagated over time. The ability to choose different twisting functions introduces a degree of freedom into the design of the particle algorithm, leading naturally to questions of optimality. In concrete terms, if the twisting function is chosen well, the twisted particle filter can estimate the marginal likelihood with greater accuracy than a standard particle filter, in turn allowing more efficient estimation inference for static parameters in the state-space model.

The investigations of [1] focussed mainly on theoretical

J. Ala-Luhtala is with IndoorAtlas Ltd., Helsinki, Finland and the Department of Mathematics, Tampere University of Technology, Tampere, Finland, e-mail: juha.ala-luhtala@tut.fi.

N. Whiteley is with the School of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, UK, e-mail: nick.whiteley@bristol.ac.uk

K. Heine is with the Department of Statistical Science, University College London, Gower Street, London, WC1E 6BT, UK, e-mail: k.heine@ucl.ac.uk

R. Piché is with the Department of Automation Science and Engineering, Tampere University of Technology, Tampere, Finland, e-mail: robert.piche@tut.fi.

analysis of twisted particle filters, studying their asymptotic properties in the regimes where the number of particles tends to infinity and where the length of the time horizon grows, under probabilistic assumptions on the observation sequence and strong regularity conditions on the statistical model.

The objectives of this paper are to present new twisted particle filtering methodology, validate it theoretically, and demonstrate its application and effectiveness within PMCMC for inferring the parameters of state-space models. Our main contributions are as follows.

1) Algorithms: We introduce a general formulation of twisted particle filters. The first novel aspect of this formulation beyond that given in [1], is that it allows various resampling methods to be incorporated in twisted particle filters. In particular, we derive a twisted particle filter around the popular systematic resampling method, which is known to reduce variance within the particle filter. The second novel aspect of the new formulation is that it allows for twisting functions which depend on historical particle states, which is an important factor when designing them efficiently in practice. The methodology of [1], which treated only multinomial resampling and twisting functions which depend only on current particle states. The utility of these algorithmic developments is that they allow for more accurate estimation of marginal likelihoods.

2) Theory: We provide novel theoretical results which justify the new algorithms and characterize their optimal operation. The first result, Theorem 1, establishes the lack-of-bias property of the marginal likelihood approximation delivered by the new twisted particle filter. The importance of this result is that it justifies the use of the twisted particle filter within PMCMC, whilst allowing for more general structure of the resampling technique and twisting function than in [1].

The second result, Theorem 2, identifies the twisting functions which are optimal for approximating the marginal likelihood for a given finite number of observations. This provides a different perspective to the results of [1], which are asymptotic in nature, considering optimality in terms of minimal variance growth rate in the regime where the length of the data record tends to infinity. Theorem 2 relies on only mild regularity assumptions on the ingredients of the twisted particle filter, whereas the results of [1] assume a particularly strong form of geometric ergodicity of the signal in the hidden Markov model (HMM) and certain uniform upper and lower bounds on the likelihood functions. Moreover, compared to the analyses of [1], the proof of Theorem 2 is less intricate, and gives the reader a more accessible route to understanding how twisted particle filters work.

3) Approximation techniques: Informed by Theorem 2, we propose methods to approximate the optimal twisting function for nonlinear Gaussian state-space models, based on ideas of Kalman filtering methodology together with local linearization using historical particle states.

4) Applications and numerical results: We provide numerical results in the context of two applications.

The first application is a range and bearing tracking problem. This is a classical nonlinear tracking scenario and serves as a benchmark application of particle filters [15], [2]. The purpose of this example is to compare the performance of twisted particle filters and the corresponding PMCMC algorithms to standard particle filters in a situation where the ground truth for static parameters is available, with simulated data. The twisted particle filters we consider employ linearization techniques to approximate the optimal twisting functions. The results we obtain illustrate that twisted particle filters can more reliably approximate marginal likelihoods for the same or less computational cost than standard particle filters. The benefits of using twisted particle filters within PMCMC are also demonstrated in terms of lower auto-correlation, and consequently more accurate approximation of posterior distributions over static parameters. We also compare tracking performance based on estimated parameter values.

The second application is a more complex indoor positioning problem. In this application a state-space model represents the unknown position of a user over time, observed indirectly and with uncertainty through received signal strength (RSS) measurements. Such data are widely available from many different wireless communication systems including mobile networks and WLAN. They have been proposed for use in location estimation in a variety of novel location-aware applications, such as environmental and structure monitoring, and many military and public safety applications, see [16], [17] and references therein. We work with a real Bluetooth RSS data set. A key task when dealing with RSS measurements is to calibrate the model by estimating unknown parameters which describe attenuation characteristics of the environment in which the user moves, since otherwise one must resort to oversimplified models [17], which exhibit inferior tracking performance.

A variety of approaches to estimating these parameters have been suggested, involving least squares [17] and weighted least squares [18] methods. These techniques provide point estimates of parameter values from a batch of data. Bayesian approaches, e.g., [19], allow additionally for uncertainty associated with estimates to be reported, and incorporate prior distributions to quantify expert knowledge and physical constraints on parameter values. They also naturally handle uncertainty over state variables when inferring parameters through marginalization.

The price to pay for the Bayesian approach is the computational cost of Monte Carlo sampling, and so our numerical investigations largely focus on computational efficiency. We compare the performance of twisted particle filters to more standard particle filters using a variety of proposal and resampling techniques. We demonstrate improved CPU-time efficiency in estimating marginal likelihoods, and we show that this efficiency is carried over to the particle MCMC algorithm, giving performance gains in terms of quality of the resulting MCMC chain compared to PMCMC using a standard particle filter. We also demonstrate favourable tracking performance using parameter estimates obtained from PMCMC.

The structure of the paper is as follows. Section II gives the problem formulation. Section III introduces a PMCMC algorithm and a standard particle filter. Section IV presents the twisted particle filtering methodology and Theorems 1-2, which characterize the lack-of-bias property and optimal

twisting functions. Computational complexity is also discussed. Section V introduces methods for approximating the optimal twisting functions in nonlinear state-space models with Gaussian noise. Section VI contains applications and numerical results. Finally, conclusions are presented in Section VII.

II. Problem formulation

We first introduce some notation. Uppercase is used to denote random variables (e.g. $X, Y, \ldots$) and realized values are denoted with lowercase (e.g. $x, y, \ldots$). For any sequence $(a_n)_{n\geq 0}$ and $s \leq k$ we write $a_{s:k} := (a_s, \ldots, a_k)$.

We consider state-space models of the form

$$\begin{aligned} X_0 \sim \mu_{0,\theta}(\cdot), \quad X_k \sim f_{k,\theta}(\cdot \,|\, X_{k-1}), \quad k \geq 1, \\ Y_k \sim g_{k,\theta}(\cdot \,|\, X_k), \quad k \geq 0, \end{aligned} \tag{1}$$

where $X_k \in \mathbb{X}$ is the state vector, $Y_k \in \mathbb{Y}_k$ is the measurement vector, $\mu_{0,\theta}(\cdot)$ is the initial distribution, $f_{k,\theta}(\cdot \,|\, x_{k-1})$ describes the transitions of the state process and $g_{k,\theta}(\cdot \,|\, x_k)$ is the conditional distribution for the measurement. All the model distributions are assumed to admit probability densities denoted with the same letter as the distribution. The joint density of the state-variables and measurements for $k \geq 1$ is given by

$$\begin{aligned} p_\theta(y_{0:k}, x_{0:k}) = \mu_{0,\theta}(x_0) g_{0,\theta}(y_0 \,|\, x_0) \\ \cdot \prod_{s=1}^{k} f_{s,\theta}(x_s \,|\, x_{s-1}) g_{s,\theta}(y_s \,|\, x_s). \end{aligned} \tag{2}$$

The parameter vector $\theta \in \mathbb{R}^{d_\theta}$ contains all the unknown parameters of the model.

We are mainly concerned in estimating the unknown parameters θ using a set of realized measurements $y_{0:t}$. In the Bayesian framework, the parameters are considered as random variables and estimates are computed using the posterior distribution

$$p(\theta \,|\, y_{0:t}) \propto p_\theta(y_{0:t}) p(\theta), \tag{3}$$

where $p_\theta(y_{0:t})$ is the likelihood and $p(\theta)$ is the prior.

With the shorthand

$$\pi^-_{k,\theta}(dx_k) := p_\theta(dx_k \,|\, y_{0:k-1}), \;\; \pi_{k,\theta}(dx_k) := p_\theta(dx_k \,|\, y_{0:k}),$$

the likelihood term can be evaluated recursively, for $k \geq 1$,

$$p_\theta(y_{0:k}) = p_\theta(y_{0:k-1}) \int_{\mathbb{X}} g_{k,\theta}(y_k \,|\, x_k) \pi^-_{k,\theta}(dx_k), \tag{4}$$

$p_\theta(y_0) = \int_{\mathbb{X}} g_{0,\theta}(y_0 \,|\, x_0) \mu_{0,\theta}(dx_0)$, and

$$\pi^-_{k,\theta}(x_k) = \int_{\mathbb{X}} f_{k,\theta}(x_k \,|\, x_{k-1}) \pi_{k-1,\theta}(dx_{k-1}), \; k \geq 1, \tag{5}$$

$$\pi_{k,\theta}(x_k) \propto \begin{cases} g_{0,\theta}(y_0 \,|\, x_0) \mu_{0,\theta}(x_0), & k = 0, \\ g_{k,\theta}(y_k \,|\, x_k) \pi^-_{k,\theta}(x_k), & k \geq 1. \end{cases} \tag{6}$$

Exact inference using (3) directly is usually intractable, since the likelihood term can be evaluated exactly for only some special models (e.g. linear Gaussian model). We consider particle filtering methods for computing unbiased estimates for the likelihood term. These can then be used as a part of particle MCMC methods that draw samples from the posterior distribution of interest.

1: Sample $\theta^0 \sim p(\theta)$
2: Obtain an unbiased estimate Z^0 of $p_{\theta^0}(y_{0:t})$
3: **for** $i \geq 1$ **do**
4: Sample $\theta^* \sim \kappa(\cdot \,|\, \theta^{i-1})$
5: Obtain an unbiased estimate Z^* of $p_{\theta^*}(y_{0:t})$
6: Set $\alpha = \min\left\{1, \frac{Z^* p(\theta^*)}{Z^{i-1} p(\theta^{i-1})} \frac{\kappa(\theta^{i-1} \,|\, \theta^*)}{\kappa(\theta^* \,|\, \theta^{i-1})}\right\}$
7: Sample U from a uniform distribution on $[0, 1]$
8: **if** $U < \alpha$ **then**
9: Set $\theta^i = \theta^*$ and $Z^i = Z^*$
10: **else**
11: Set $\theta^i = \theta^{i-1}$ and $Z^i = Z^{i-1}$
12: **end if**
13: **end for**

Algorithm 1. Particle marginal Metropolis-Hastings

III. Particle MCMC

In this section we describe methods for drawing samples from the parameter posterior distribution in (3). Algorithms targeting only the parameter posterior are often called marginal algorithms, because samples are drawn only from the marginal posterior $p(\theta \,|\, y_{0:t})$ instead of the full posterior $p(x_{0:t}, \theta \,|\, y_{0:t})$.

MCMC methods generate samples from the target posterior distribution by simulating a Markov chain $\theta^0, \theta^1, \ldots$ that has the target posterior distribution as a stationary distribution [7]. One of the best known and general MCMC methods is the Metropolis-Hastings (MH) algorithm, where a new sample θ^* at step i is generated from a proposal distribution $\kappa(\cdot \,|\, \theta^{i-1})$. The generated sample θ^* is then accepted with probability

$$\min\left\{1, \frac{p_{\theta^*}(y_{0:t}) p(\theta^*)}{p_{\theta^{i-1}}(y_{0:t}) p(\theta^{i-1})} \frac{\kappa(\theta^{i-1} \,|\, \theta^*)}{\kappa(\theta^* \,|\, \theta^{i-1})}\right\}. \tag{7}$$

To compute this acceptance probability, we need to evaluate likelihood terms $p_\theta(y_{0:t})$, but that is not possible for a general nonlinear state-space model. However, if an unbiased estimator for the likelihood is available, it is still possible to construct an MCMC algorithm to sample from the posterior distribution [8], [9]. For state-space models, we can use particle filters as unbiased estimators of the likelihood [11]. A Metropolis-Hastings algorithm using particle filters to estimate the likelihood terms, called particle marginal Metropolis-Hastings (PMMH) [11], is given in Algorithm 1.

A. Particle filtering

We proceed with an account of a standard particle filter. Our notation is in some places a little non-standard, but is chosen deliberately to help with the derivation of twisted particle filters in Section IV. Henceforth, for notational simplicity, we often omit the subscript θ and implicitly assume that the distributions can depend on the parameters.

We denote the set of $n \geq 1$ particles at time $k \geq 0$ by $\xi_k = (\xi_k^i)_{i=1}^n$, with corresponding unnormalized weights $W_k = (W_k^i)_{i=1}^n$. The filtering distribution is approximated by

$$\pi_{k,\theta}(dx_k) \approx \frac{\sum_{i=1}^n W_k^i \delta_{\xi_k^i}(dx_k)}{\sum_{i=1}^n W_k^i}, \tag{8}$$

1: **for** $1 \leq i \leq n$ **do**
2: Sample $\xi_0^i \sim q_0(\cdot)$
3: Set $W_0^i = g_0(y_0 \,|\, \xi_0^i)\mu_0(\xi_0^i)/q_0(\xi_0^i)$
4: **end for**
5: Set $Z_0 = \frac{1}{n}\sum_{i=1}^n W_0^i$
6: **for** $1 \leq k \leq t$ **do**
7: Sample $U_{k-1} \sim \mathcal{U}[0,1]^m$
8: Set $A_{k-1} = r(U_{k-1}, W_{k-1})$
9: **for** $1 \leq i \leq n$ **do**
10: Sample $\xi_k^i \sim q_k(\cdot \,|\, \mathscr{L}_{k-1}^{A_{k-1}^i})$
11: Set $W_k^i = \dfrac{g_k(y_k \,|\, \xi_k^i) f_k(\xi_k^i \,|\, \xi_{k-1}^{A_{k-1}^i})}{q_k(\xi_k^i \,|\, \mathscr{L}_{k-1}^{A_{k-1}^i})}$
12: **end for**
13: Set $Z_k = Z_{k-1}\frac{1}{n}\sum_{i=1}^n W_k^i$
14: **end for**

Algorithm 2. Particle filter

where $\delta_{\xi_k^i}(\cdot)$ denotes a unit point mass centered at ξ_k^i.

In order to describe the sampling mechanism for the particles and understand certain properties of the algorithm it is convenient to also introduce, for each $k \geq 0$, the ancestor indicator variables $A_k = (A_k^i)_{i=1}^n$, where each A_k^i takes a value in $\{1, \ldots, n\}$. If we also define for each $k \geq 0$ and $i \in \{1, \ldots, n\}$, $(B_{k,j}^i)_{j=0}^k$ by letting $B_{k,k}^i := i$ and for $k > 0$, recursively $B_{k,j} := A_j^{B_{k,j+1}}$, $j = k-1, \ldots, 0$, then we can write the "ancestral line" of particle ξ_k^i as

$$\mathscr{L}_k^i := (\xi_k^i, \xi_{k-1}^{B_{k,k-1}^i}, \ldots, \xi_0^{B_{k,0}^i}), \tag{9}$$

which is a $\mathbb{X}^{k+1}$-valued random variable.

A particle filter is given in Algorithm 2. Here the proposal distributions $(q_k)_{k\geq 0}$ are assumed to be chosen such that for each $k \geq 0$ the weights W_k are strictly positive and finite. Each q_k may be chosen to depend also on the observations $y_{0:k}$, but this dependence is suppressed from the notation.

B. Resampling

Lines 7 and 8 in Algorithm 2 together implement a generic resampling operation. Line 7 generates $U_{k-1} = (U_{k-1}^i)_{i=1}^m$ consisting of $m \geq 1$ i.i.d. random variables, each uniformly distributed on $[0,1]$. Line 8 passes U_{k-1} and the unnormalized weights W_{k-1} to a deterministic mapping $r\colon [0,1]^m \times \mathbb{R}_+^n \to \{1, \ldots, n\}^n$, which returns the ancestor indicator variables $A_{k-1} = (A_{k-1}^i)_{i=1}^n$. With $r^i(U_{k-1}, W_{k-1})$ indicating the ith element in the vector returned by r, for brevity we sometimes write $r_{k-1}^i(U_{k-1}) \equiv r^i(U_{k-1}, W_{k-1})$.

A variety of resampling mechanisms can be cast in this form through specific choices of m and r. We describe here two well known schemes: the multinomial and systematic methods; see [20] for background information. These techniques are standard; the details are included here in order to prepare for the presentation of the non-standard resampling techniques in twisted particle filters.

1) **Multinomial resampling**: We have $m = n$ and the mapping r is defined as

$$r^i(u,w) = j \quad \Leftrightarrow \quad u^i \in (d^{j-1}, d^j], \tag{10}$$

where $d^0 = 0$ and $d^i = \sum_{j=1}^i w^j / (\sum_{j=1}^n w^j)$.

2) **Systematic resampling**: We have $m = 1$ and the mapping r is defined as

$$r^i(u,w) = j \quad \Leftrightarrow \quad u + i - 1 \in (nd^{j-1}, nd^j], \tag{11}$$

where $d^0 = 0$ and $d^i = \sum_{j=1}^i w^j / (\sum_{j=1}^n w^j)$.

Systematic resampling is computationally light and has been found to have good empirical performance, although theoretical analysis is difficult due to high dependence between the resampled particles. Nevertheless, it is known, see e.g. [20], that both multinomial and systematic resampling satisfy Assumption 1 below.

We define the shorthand notation $\mathscr{F}_0 := \xi_0$ and for $k \geq 1$, $\mathscr{F}_k := (\xi_0, U_0, \xi_1, \ldots, U_{k-1}, \xi_k)$.

Assumption 1. *The mapping r is such that for any $k \geq 0$ and integrable function $\varphi : \mathbb{X}^{k+1} \to \mathbb{R}$,*

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \varphi(\mathscr{L}_k^{r_k^i(U_k)}) \,\middle|\, \mathscr{F}_k\right] = \frac{\sum_{i=1}^n W_k^i \varphi(\mathscr{L}_k^i)}{\sum_{i=1}^n W_k^i},$$

where $\mathbb{E}$ denotes expectation when sampling according to Algorithm 2.

Lines 5 and 13 compute a sequence $(Z_k)_{k=0}^t$, where each Z_k is an estimate of $p(y_{0:k})$. The following proposition justifies the use of Algorithm 2 to provide an unbiased estimate of $p(y_{0:t})$ at line 5 of Algorithm 1. This kind of result is well known; a proof is outlined in Appendix A for completeness.

Proposition 1. *If Assumption 1 holds, then for each $k \geq 0$, $\mathbb{E}[Z_k] = p(y_{0:k})$.*

IV. Twisted particle filters

In order to introduce and validate twisted particle filters we think more explicitly about ξ_0 and the sequence $(\xi_k, U_{k-1})_{k\geq 1}$ as a stochastic process and consider the following initial and conditional distributions, according to which ξ_0 and $(\xi_k, U_{k-1})_{k\geq 1}$ evolve when sampled through Algorithm 2.

$$\mathbf{M}_0(d\xi_0) = \prod_{i=1}^n q_0(d\xi_0^i), \tag{12a}$$

$$\mathbf{M}_k(d\xi_k, du_{k-1} \,|\, \mathscr{F}_{k-1}) = \mathcal{U}(du_{k-1}) \prod_{i=1}^n q_k(d\xi_k^i \,|\, \mathscr{L}_{k-1}^{r_{k-1}^i(u_{k-1})}), \tag{12b}$$

where $\mathcal{U}(du)$ denotes the uniform distribution on $[0,1]^m$.

Twisted particle filters are obtained by sampling the process ξ_0, $(\xi_k, U_{k-1})_{k\geq 1}$ from alternatives to (12a)–(12b), which we discuss in more detail below.

Remark 1. *For historical perspective, we note that the idea of constructing alternative distributions over the random variables in particle filters appears in some of the theoretical arguments which justify PMCMC [11]. However, the specifics of twisted particle filters are more akin to eigenfunction changes of measure for branching processes, which were studied earlier in the stochastic processes literature, see [21, Section 3] and references therein.*

Let $(\psi_k)_{k\geq 0}$ be a sequence of strictly positive functions, such that $\psi_0 \colon \mathbb{X} \to \mathbb{R}_+$ and for $k \geq 1$, $\psi_k \colon \mathbb{X}^{k+1} \to \mathbb{R}_+$. We shall often write interchangeably $\psi_k(x_{0:k-1}, x_k) \equiv \psi_k(x_{0:k})$. Each ψ_k may also depend on $\mathscr{F}_{k-1}$ and any number of the measurements y_k, but this dependence is suppressed from the notation.

The initial and conditional distributions for the twisted particle filter are given by

$$\widetilde{\mathbf{M}}_0(d\xi_0) \propto \frac{1}{n}\sum_{s=1}^{n} \mathbf{M}_0(d\xi_0)\psi_0(\xi_0^s), \tag{13a}$$

$$\widetilde{\mathbf{M}}_k(d\xi_k, du_{k-1} \,|\, \mathscr{F}_{k-1}) \propto \frac{1}{n}\sum_{s=1}^{n} \mathbf{M}_k(d\xi_k, du_{k-1} \,|\, \mathscr{F}_{k-1})\psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, \xi_k^s), \tag{13b}$$

where the functions ψ_k are called "twisting functions". To avoid some tangential complications we shall assume henceforth that $\sup_x \psi_k(x) < \infty$ for each $k \geq 0$, which is sufficient to ensure that the integrals needed to normalize $\widetilde{\mathbf{M}}_0$ and each $\widetilde{\mathbf{M}}_k$ are finite.

A more explicit expression for $\widetilde{\mathbf{M}}_0$ is obtained by plugging in (12a) and normalizing, to give

$$\widetilde{\mathbf{M}}_0(d\xi_0) = \frac{1}{n}\sum_{s=1}^{n} \widetilde{q}_0(d\xi_0^s)\prod_{i\neq s} q_0(d\xi_0^i),$$

where $\widetilde{q}_0(d\xi_0^s) := \psi_0(\xi_0)q_0(d\xi_0^s)/\int \psi_0(x)q_0(dx)$. So to sample from $\widetilde{\mathbf{M}}_0$, one first draws a random variable, say S_0, from the uniform distribution on $\{1,\dots,n\}$, then samples $\xi_0^{S_0} \sim \widetilde{q}_0(\cdot)$ and $\xi_0^i \sim q_0(\cdot)$ for $i \neq S_0$. Deriving a similar sampling recipe for $\widetilde{\mathbf{M}}_k$ is somewhat more involved. We state the resulting procedure in Algorithm 3, then formalize its validity and other properties in Theorems 1 and 2.

To write out Algorithm 3 we need a few more definitions. For $k \geq 0$, define the twisted (unnormalized) weights

$$\widetilde{W}_k^i := W_k^i \widetilde{V}_k^i, \quad 1 \leq i \leq n, \tag{14}$$

where

$$\widetilde{V}_k^i := \int_{\mathbb{X}} \psi_{k+1}(\mathscr{L}_k^i, x_{k+1}) q_{k+1}(dx_{k+1} \,|\, \mathscr{L}_k^i). \tag{15}$$

For $k \geq 1$, define the twisted proposal distribution

$$\widetilde{q}_k(dx_k \,|\, x_{0:k-1}) \propto \psi_k(x_{0:k}) q_k(dx_k \,|\, x_{0:k-1}). \tag{16}$$

Let S_k be a discrete random variable conditional on $(\mathscr{L}_{k-1}^i)_{i=1}^n$, having distribution $\widetilde{\mathcal{S}}_k(\cdot)$ on $\{1,\dots,n\}$, whose probabilities are proportional to

$$\widetilde{\mathcal{S}}_k(S_k = s) \propto \int_{[0,1]^m} \mathcal{U}(du) \cdot \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u)}, x_k) q_k(dx_k \,|\, \mathscr{L}_{k-1}^{r_{k-1}^s(u)}). \tag{17}$$

Also introduce a distribution $\widetilde{\mathcal{U}}_{k-1}(\cdot \,|\, s)$ on $[0,1]^m$ given by

$$\widetilde{\mathcal{U}}_{k-1}(du \,|\, s) \propto \mathcal{U}(du) \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u)}, x_k) q_k(dx_k \,|\, \mathscr{L}_{k-1}^{r_{k-1}^s(u)}). \tag{18}$$

Sample S_0 uniformly from $\{1,\dots,n\}$
Sample $\xi_0^{S_0} \sim \widetilde{q}_0(\cdot)$
for $i \neq S_0$ **do**
  Sample $\xi_0^i \sim q_0(\cdot)$
end for
for $1 \leq i \leq n$ **do**
  Set $W_0^i = g_0(y_0 \,|\, \xi_0^i)\mu_0(\xi_0^i)/q_0(\xi_0^i)$
end for
Set $\widetilde{Z}_0 = \dfrac{\sum_{i=1}^n W_0^i \int_{\mathbb{X}} \psi_0(x_0) q_0(dx_0)}{\sum_{i=1}^n \psi_0(\xi_0^i)}$
for $1 \leq k \leq t$ **do**
  Sample $S_k \sim \widetilde{\mathcal{S}}_k(\cdot)$
  Sample $U_{k-1} \sim \widetilde{\mathcal{U}}_{k-1}(\cdot \,|\, S_k)$
  Set $A_{k-1} = r(U_{k-1}, W_{k-1})$
  Sample $\xi_k^{S_k} \sim \widetilde{q}_k(\cdot \,|\, \mathscr{L}_{k-1}^{A_{k-1}^{S_k}})$
  for $i \neq S_k$ **do**
    Sample $\xi_k^i \sim q_k(\cdot \,|\, \mathscr{L}_{k-1}^{A_{k-1}^i})$
  end for
  for $1 \leq i \leq n$ **do**
    Set $W_k^i = \dfrac{g_k(y_k \,|\, \xi_k^i) f_k(\xi_k^i \,|\, \xi_{k-1}^{A_{k-1}^i})}{q_k(\xi_k^i \,|\, \mathscr{L}_{k-1}^{A_{k-1}^i})}$
    Set $\widetilde{V}_{k-1}^i = \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^i, x) q_k(dx \,|\, \mathscr{L}_{k-1}^i)$
    Set $\widetilde{W}_{k-1}^i = W_{k-1}^i \widetilde{V}_{k-1}^i$
  end for
  Set $\widetilde{Z}_k = \widetilde{Z}_{k-1} \dfrac{\sum_{i=1}^n W_k^i}{\sum_{i=1}^n W_{k-1}^i} \dfrac{\sum_{i=1}^n \widetilde{W}_{k-1}^i}{\sum_{i=1}^n \psi_k(\mathscr{L}_k^i)}$
end for

Algorithm 3. Twisted particle filter

Note that the distributions $\widetilde{\mathcal{S}}_k$ and $\widetilde{\mathcal{U}}_{k-1}$ depend on the resampling method defined through the mapping r. Details of how to sample from these distributions in the cases when r corresponds to multinomial or systematic resampling are given in Sections IV-A and IV-B.

Our first main result, Theorem 1, establishes that Algorithm 3 indeed samples from (13a)–(13b) and delivers unbiased estimates of $p(y_{0:k})$, which justifies its use within Algorithm 1. The proof is given in Appendix B.

Theorem 1. *The random variables ξ_0 and $(\xi_k, U_{k-1})_{k\geq 1}$ sampled using Algorithm 3 are drawn from (13a)–(13b). Furthermore, if Assumption 1 holds, then for each $k \geq 0$,*

$$\widetilde{\mathbb{E}}[\widetilde{Z}_k] = \mathbb{E}[Z_k] = p(y_{0:k}), \tag{19}$$

where $\widetilde{\mathbb{E}}$ (resp. $\mathbb{E}$) denotes expectation w.r.t. (13a)–(13b) *(resp.* (12a)–(12b)*).*

Theorem 2 identifies the choice of the functions $(\psi_k)_{0\leq k\leq t}$ which are ideal for estimating $p(y_{0:t})$. The proof is given in Appendix C.

Theorem 2. *If we choose*

$$\psi_k(x_{0:k}) = \begin{cases} \dfrac{\mu_0(x_0)p(y_{0:t} \mid x_0)}{q_0(x_0)}, & k = 0, \\ \dfrac{f_k(x_k \mid x_{k-1})p(y_{k:t} \mid x_k)}{q_k(x_k \mid x_{0:k-1})}, & 1 \leq k \leq t, \end{cases} \tag{20}$$

then $\widetilde{Z}_t = p(y_{0:t})$.

The choice of ψ_k identified in (20) is of course not usually available in practice, but Theorem 2 motivates us to consider twisting functions of the form

$$\psi_{k,l}(x_{0:k}) = \begin{cases} \dfrac{\mu_0(x_0)\phi_{0,l}(x_0)}{q_0(x_0)}, & k = 0, \\ \dfrac{f_k(x_k \mid x_{k-1})\phi_{k,l}(x_{0:k})}{q_k(x_k \mid x_{0:k-1})}, & 1 \leq k \leq t, \end{cases} \tag{21}$$

where the functions $\phi_{k,l} \colon \mathbb{X}^{k+1} \to [0,1]$ are chosen to approximate $p(y_{k:k+l} \mid x_k)$, possibly also depending on $\mathscr{F}_{k-1}$, and l is a positive integer parameter such that $0 \leq l \leq t-k$, which specifies how many future measurements are used in the twisting function. Devising such approximations is the subject of Section V.

We conclude Section IV by showing how to sample S_k and U_{k-1} on lines 11 and 12 in Algorithm 3.

A. Twisted multinomial resampling

In this case $m = n$, and using definition (10) for r_{k-1}, it is easily checked that the probabilities $\widetilde{\mathcal{S}}_k(S_k = s)$ in (17) are independent of the value s, i.e. $\widetilde{\mathcal{S}}_k$ is the uniform distribution over $\{1, \ldots, n\}$.

The density function corresponding to (18) can be written as

$$\begin{aligned} &\widetilde{\mathcal{U}}_{k-1}(u \mid s) \\ &\propto \mathbb{I}_{[0,1]}(u^s) \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u^s)}, x_k) q_k(dx_k \mid \mathscr{L}_{k-1}^{r_{k-1}^s(u^s)}) \\ &\quad \cdot \prod_{i \neq s} \mathbb{I}_{[0,1]}(u^i) \\ &= \left[\sum_{j=1}^{n} \mathbb{I}_{(d^{j-1}, d^j]}(u^s) \tilde{v}_{k-1}^j \right] \prod_{i \neq s} \mathbb{I}_{[0,1]}(u^i), \end{aligned}$$

where the equality uses (10), and $d_{k-1}^0 = 0$, $d_{k-1}^j = \sum_{i=1}^{j} w_{k-1}^i / \sum_{i=1}^{n} w_{k-1}^i$ for $1 \leq j \leq n$, for any set $\mathcal{I}$, $\mathbb{I}_{\mathcal{I}}(u) = 1$, when $u \in \mathcal{I}$ and zero otherwise, and the terms $\tilde{v}_{k-1}^j$ are given by (15).

We therefore have the following procedure for sampling S_k and U_{k-1} from $\widetilde{\mathcal{S}}_k(\cdot)$ and $\widetilde{\mathcal{U}}_{k-1}(\cdot \mid s)$ respectively:

1) Sample S_k uniformly from $\{1, \ldots, n\}$
2) Sample index J_{k-1} from the discrete distribution on $\{1, \ldots, n\}$ such that the probability that $J_{k-1} = j$ is proportional to

$$\int_{[0,1]} \mathbb{I}_{(d_{k-1}^{j-1}, d_{k-1}^j]}(u^s)\, du^s\, \tilde{v}_{k-1}^j = w_{k-1}^j \tilde{v}_{k-1}^j$$

3) Sample $U_{k-1}^{S_k}$ from the uniform distribution on $(d_{k-1}^{J_{k-1}-1}, d_{k-1}^{J_{k-1}}]$ and for each $i \neq S_k$, U_{k-1}^i from the uniform distribution on $[0,1]$

B. Twisted systematic resampling

In this case we have $m = 1$, and using definition (11) for r_{k-1}, the probabilities in (17) are

$$\begin{aligned} &\widetilde{\mathcal{S}}_k(S_k = s) \\ &\propto \int_{[0,1]} \mathcal{U}(du) \int_{\mathbb{X}} q_k(dx_k \mid \mathscr{L}_{k-1}^{r_{k-1}^s(u)}) \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u)}, x_k) \\ &= \sum_{j=1}^{n} \int_{[0,1]} \mathbb{I}_{\mathcal{I}_{k-1}^{s,j}}(u)\, du \int_{\mathbb{X}} q_k(dx_k \mid \mathscr{L}_{k-1}^{j}) \psi_k(\mathscr{L}_{k-1}^{j}, x_k) \\ &= \sum_{\{j \mid \mathcal{I}_{k-1}^{j,s} \neq \emptyset\}} \Big[\min(n d_{k-1}^j - s + 1, 1) \\ &\qquad - \max(n d_{k-1}^{j-1} - s + 1, 0) \Big] \tilde{v}_{k-1}^j, \end{aligned} \tag{22}$$

where the first equality follows from (11), and $\mathcal{I}_{k-1}^{s,j} = (n d_{k-1}^{j-1} - s + 1, n d_{k-1}^j - s + 1] \cap [0,1]$ and $(d_{k-1}^j)_{j=0}^n$ are defined as in the twisted multinomial resampling.

The probability density function corresponding to (18) can be written as

$$\begin{aligned} &\widetilde{\mathcal{U}}_{k-1}(u \mid s) \\ &\propto \mathbb{I}_{[0,1]}(u) \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u)}, x_k) q_k(dx_k \mid \mathscr{L}_{k-1}^{r_{k-1}^s(u)}) \\ &= \sum_{j=1}^{n} \mathbb{I}_{\mathcal{I}_{k-1}^{s,j}}(u) \tilde{v}_{k-1}^j, \end{aligned}$$

where the equality follows from (11).

This leads to the following procedure for sampling S_k and U_{k-1} from $\widetilde{\mathcal{S}}_k(\cdot)$ and $\widetilde{\mathcal{U}}_{k-1}(\cdot \mid s)$ respectively:

1) Sample S_k from a distribution over $\{1, \ldots, n\}$ with probabilities given by (22)
2) Sample index J_{k-1} from the discrete distribution on $\{1, \ldots, n\}$ such that the probability that $J_{k-1} = j$ is proportional to

$$\begin{aligned} &\int_{[0,1]} \mathbb{I}_{\mathcal{I}_{k-1}^{S_k,j}}(u)\, du\, \tilde{v}_{k-1}^j \\ &= \Big[\min(n d_{k-1}^j - S_k + 1, 1) \\ &\qquad - \max(n d_{k-1}^{j-1} - S_k + 1, 0) \Big] \tilde{v}_{k-1}^j, \end{aligned}$$

   if $\mathcal{I}_{k-1}^{S_k,j} \neq \emptyset$, and otherwise the probability that $J_{k-1} = j$ is zero.
3) Sample U_{k-1} from the uniform distribution on $\mathcal{I}_{k-1}^{S_k,J_{k-1}}$

C. Complexity of twisted resampling methods

Twisted multinomial resampling involves sampling 2 times from a discrete distribution with n elements and n times from a continuous uniform distribution, and can be implemented in $\mathcal{O}(n)$ time.

Twisted systematic resampling involves sampling two times from a discrete distribution with n elements and one time from

a continuous uniform distribution, and can be implemented in $\mathcal{O}(n)$ time. Compared to twisted multinomial resampling, some computation time is saved since only one draw from the continuous uniform distribution is needed. However, the computation of the probabilities for the discrete distributions is computationally more involved for the twisted systematic resampling.

The overall complexity of Algorithm 3 depends on the specific nature of the twisting function and how it is computed. This is a problem-specific issue, which we discuss in the context of a particular family of models and twisting functions in Section V-C.

V. Twisted particle filters for Gaussian state-space models

In this section, we present methods for approximating the optimal twisting function in Gaussian state-space models with $\mathbb{X} = \mathbb{R}^{d_x}$, $\mathbb{Y} = \mathbb{R}^{d_y}$ and

$$\mu_0(\cdot) = \mathcal{N}(\cdot \,|\, \nu_0, \mathbf{P}_0), \tag{23a}$$
$$f_k(\cdot \,|\, x_{k-1}) = \mathcal{N}(\cdot \,|\, c_{k-1}(x_{k-1}), \mathbf{Q}_{k-1}), \quad k \geq 1, \tag{23b}$$
$$g_k(\cdot \,|\, x_k) = \mathcal{N}(\cdot \,|\, h_k(x_k), \mathbf{R}_k), \quad k \geq 0, \tag{23c}$$

where $\mathcal{N}(\cdot \,|\, \nu, \mathbf{P})$ denotes a Gaussian distribution with mean vector ν and covariance matrix $\mathbf{P}$. The mean functions $c_{k-1}(x_{k-1})$ and $h_k(x_k)$ can be nonlinear functions of the state vector.

To use the twisted particle filter in practice, we need to evaluate the integrals in (15) and sample from the twisted distributions given by (16). For the Gaussian model, we choose an exponential form for the function $\phi_{k,l}$ in (21), given by

$$\phi_{k,l}(x_{0:k}) = \alpha_{k,l} \exp\left\{ -\frac{1}{2} x_k^T \mathbf{\Gamma}_{k,l} x_k + x_k^T \beta_{k,l} \right\}, \tag{24}$$

where $\alpha_{k,l} \equiv \alpha_{k,l}(x_{0:k-1}) \in \mathbb{R}^+$, $\beta_{k,l} \equiv \beta_{k,l}(x_{0:k-1}) \in \mathbb{R}^{d_x}$ and $\mathbf{\Gamma}_{k,l} \equiv \mathbf{\Gamma}_{k,l}(x_{0:k-1}) \in \mathbb{R}^{d_x \times d_x}$ are parameters, possibly depending on $\mathscr{F}_{k-1}$ and any number of measurements. For $k \geq 1$, we use shorthand notation $\alpha_{k,l}^i = \alpha_{k,l}(\mathscr{L}_{k-1}^i)$, $\beta_{k,l}^i = \beta_{k,l}(\mathscr{L}_{k-1}^i)$ and $\mathbf{\Gamma}_{k,l}^i = \mathbf{\Gamma}_{k,l}(\mathscr{L}_{k-1}^i)$. Methods for computing these parameters are considered in Sections V-A and V-B.

With twisting function given by (21) and (24), we have $\widetilde{q}_0(\cdot) = \mathcal{N}(\cdot \,|\, \mu_{0,l}, \mathbf{\Sigma}_{0,l})$, where

$$\mu_{0,l} = \mathbf{\Sigma}_{0,l} \left(\mathbf{P}_0^{-1} \nu_0 + \beta_{0,l} \right), \tag{25a}$$
$$\mathbf{\Sigma}_{0,l} = \left(\mathbf{P}_0^{-1} + \mathbf{\Gamma}_{0,l} \right)^{-1}. \tag{25b}$$

For $k \geq 1$ and $1 \leq i \leq n$, we have $\widetilde{q}_k(\cdot \,|\, \mathscr{L}_{k-1}^i) = \mathcal{N}(\cdot \,|\, \mu_{k,l}^i, \mathbf{\Sigma}_{k,l}^i)$, where

$$\mu_{k,l}^i = \mathbf{\Sigma}_{k,l}^i \left(\mathbf{Q}_{k-1}^{-1} c_{k-1}(\xi_{k-1}^i) + \beta_{k,l}^i \right), \tag{26a}$$
$$\mathbf{\Sigma}_{k,l}^i = \left(\mathbf{Q}_{k-1}^{-1} + \mathbf{\Gamma}_{k,l}^i \right)^{-1}. \tag{26b}$$

The initial likelihood estimate in the twisted particle filter is now given by

$$\widetilde{Z}_0 = \left[\frac{\alpha_{0,l} |\mathbf{\Sigma}_{0,l}|^{1/2}}{|\mathbf{P}_0|^{1/2}} \frac{\exp\left\{ \frac{1}{2} \mu_{0,l}^T \mathbf{\Sigma}_{0,l}^{-1} \mu_{0,l} \right\}}{\exp\left\{ \frac{1}{2} \nu_0^T \mathbf{P}_0^{-1} \nu_0 \right\}} \right] \frac{\sum_{i=1}^n W_0^i}{\sum_{i=1}^n \psi_0(\xi_0^i)}, \tag{27}$$

where $|\mathbf{P}|$ denotes the determinant of a matrix $\mathbf{P}$. The integral in (15) can be computed to give

$$\widetilde{V}_k^i = \alpha_{k+1,l}^i \frac{|\mathbf{\Sigma}_{k+1,l}^i|^{1/2}}{|\mathbf{Q}_k|^{1/2}} \cdot \frac{\exp\left\{ \frac{1}{2} (\mu_{k+1,l}^i)^T \left(\mathbf{\Sigma}_{k+1,l}^i \right)^{-1} \mu_{k+1,l}^i \right\}}{\exp\left\{ \frac{1}{2} c_k(\xi_k^i)^T \mathbf{Q}_k^{-1} c_k(\xi_k^i) \right\}}, \tag{28}$$

for $k \geq 0$ and $1 \leq i \leq n$.

The resulting algorithm for the twisted particle filter for Gaussian state-space models is given in Algorithm 4. We conclude Section V by presenting two methods for computing the twisting function parameters on lines 1 and 13. Whilst we focus on the case of Gaussian disturbances, one could follow an almost identical approach to constructing a twisting function for a model in which the disturbances are non-Gaussian, but of known mean and covariance. In particular, one replaces respectively $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$ by the conditional covariances of $X_k|x_{k-1}$ and $Y_k|x_k$, and $c_{k-1}(x_{k-1})$ and $h_k(x_k)$ by the conditional means of $X_k|x_{k-1}$ and $Y_k|x_k$. Exponential-family disturbances could be treated with the kind of techniques explored in [22].

A. Twisting function using local linearization

For a linear Gaussian model the term $p(y_{k:k+l} \,|\, x_k)$, as a function of x_k, is exactly of the exponential form in (24). For a nonlinear Gaussian model, we can therefore compute an approximation of $p(y_{k:k+l} \,|\, x_k)$ by considering linearized transition and measurement functions.

We propose to use a local Taylor series based linearization using the extended Kalman filter (EKF). The local linearization method for computing the twisting function parameters $\alpha_{k,l}$, $\beta_{k,l}$ and $\mathbf{\Gamma}_{k,l}$ is summarized in Algorithms 5 and 6 and details are given in the following equations.

We first present the equations for computing the linearized transition functions $c_{k+s-1}(x_{k+s-1}) \approx \mathbf{C}_{k+s-1} x_{k+s-1} + \hat{c}_{k+s-1}$ and linearized measurement functions $h_{k+s}(x_{k+s}) \approx \mathbf{H}_{k+s} x_{k+s} + \hat{h}_{k+s}$ for $0 \leq s \leq l$ using the EKF local linearization.

For $k \geq 1$ and $1 \leq i \leq n$, the EKF algorithm is initialized with $\hat{x}_{k-1} = \xi_{k-1}^i$ and $\hat{\mathbf{P}}_{k-1} = \mathbf{0}$. For $0 \leq s \leq l$, we recursively compute $\mathbf{C}_{k+s-1}$, $\hat{c}_{k+s-1}$, $\mathbf{H}_{k+s}$ and $\hat{h}_{k+s}$ by first linearizing the transition function using the EKF prediction step equations

$$\hat{x}_{k+s}^- = c_{k+s-1}(\hat{x}_{k+s-1}), \tag{29a}$$
$$\mathbf{C}_{k+s-1} = \left[\frac{\partial}{\partial x} c_{k+s-1}(x) \right]_{x=\hat{x}_{k+s-1}}, \tag{29b}$$
$$\hat{c}_{k+s-1} = c_{k+s-1}(\hat{x}_{k+s-1}) - \mathbf{C}_{k+s-1} \hat{x}_{k+s-1}, \tag{29c}$$
$$\hat{\mathbf{P}}_{k+s}^- = \mathbf{C}_{k+s-1} \hat{\mathbf{P}}_{k+s-1} \mathbf{C}_{k+s-1}^T + \mathbf{Q}_{k+s-1}, \tag{29d}$$

where for a vector-valued function c, $\left[\frac{\partial}{\partial x} c(x) \right]_{x=\hat{x}}$ denotes the Jacobian matrix, evaluated at the point $x = \hat{x}$.

Set $(\alpha_{0,l}, \beta_{0,l}, \mathbf{\Gamma}_{0,l})$ using Algorithm 5 or 7
Set $\mu_{0,l}$ and $\mathbf{\Sigma}_{0,l}$ using (25a)–(25b)
Sample S_0 uniformly from $\{1, \dots, n\}$
Sample $\xi_0^{S_0} \sim \mathcal{N}(\cdot \,|\, \mu_{0,l}, \mathbf{\Sigma}_{0,l})$
for $i \neq S_0$ **do**
    Sample $\xi_0^i \sim q_0(\cdot)$
end for
for $1 \leq i \leq n$ **do**
    Set $W_0^i = g(y_0 \,|\, \xi_0^i)\mu_0(\xi_0^i)/q_0(\xi_0^i)$
end for
Set $\widetilde{Z}_0$ using (27)
for $1 \leq k \leq t$ **do**
    Set $(\alpha_{k,l}^i, \beta_{k,l}^i, \mathbf{\Gamma}_{k,l}^i)_{i=1}^n$ using Algorithm 6 or 7
    for $1 \leq i \leq n$ **do**
        Set $\mu_{k,l}^i$ and $\mathbf{\Sigma}_{k,l}^i$ using (26a)–(26b)
    end for
    Sample $S_k \sim \widetilde{\mathcal{S}}_k(\cdot)$
    Sample $U_{k-1} \sim \mathcal{U}_{k-1}(\cdot \,|\, S_k)$
    Set $A_{k-1} = r(U_{k-1}, W_{k-1})$
    Sample $\xi_k^{S_k} \sim \mathcal{N}(\cdot \,|\, \mu_{k,l}^{A_{k-1}^{S_k}}, \mathbf{\Sigma}_{k,l}^{A_{k-1}^{S_k}})$
    for $i \neq S_k$ **do**
        Sample $\xi_k^i \sim q_k(\cdot \,|\, \mathscr{L}_{k-1}^{A_{k-1}^i})$
    end for
    for $1 \leq i \leq n$ **do**
        Set $W_k^i = \dfrac{g(y_k \,|\, \xi_k^i) f(\xi_k^i \,|\, \xi_{k-1}^{A_{k-1}^i})}{q_k(\xi_k^i \,|\, \mathscr{L}_{k-1}^{A_{k-1}^i})}$
        Set $\widetilde{V}_{k-1}^i$ using (28) and $\widetilde{W}_{k-1}^i = W_{k-1}^i \widetilde{V}_{k-1}^i$
    end for
    Set $\widetilde{Z}_k = \widetilde{Z}_{k-1} \dfrac{\sum_{i=1}^n W_k^i}{\sum_{i=1}^n W_{k-1}^i} \dfrac{\sum_{i=1}^n \widetilde{W}_{k-1}^i}{\sum_{i=1}^n \psi_k(\mathscr{L}_k^i)}$
end for

Algorithm 4. Twisted particle filter for Gaussian model

Set $\hat{x}_0^- = \nu_0$ and $\hat{\mathbf{P}}_0^- = \mathbf{P}_0$
Set $\hat{x}_0$, $\hat{\mathbf{P}}_0$, $\mathbf{H}_0$ and $\hat{h}_0$ using (30)-(31)
Set $(\alpha_{0,0},\ \beta_{0,0},\ \mathbf{\Gamma}_{0,0})$ using (32)
for $1 \leq s \leq l$ **do**
    Set $\hat{x}_s$, $\hat{\mathbf{P}}_s$, $\mathbf{C}_{s-1}$, $\hat{c}_{s-1}$, $\mathbf{H}_s$ and $\hat{h}_s$ using (29)-(31)
    Set $(\alpha_{0,s},\ \beta_{0,s},\ \mathbf{\Gamma}_{0,s})$ using (33)–(35)
end for
Return $(\alpha_{0,l}, \beta_{0,l}, \mathbf{\Gamma}_{0,l})$

Algorithm 5. Twisting function parameters for $k = 0$ using EKF linearization

The linearization for the measurement function is obtained by first computing the EKF update step equations

$$\mathbf{H}_{k+s}^- = \left[\frac{\partial}{\partial x} h_{k+s}(x)\right]_{x=\hat{x}_{k+s}^-}, \tag{30a}$$

$$\mathbf{S}_{k+s}^- = \mathbf{H}_{k+s}^- \hat{\mathbf{P}}_{k+s}^- \left(\mathbf{H}_{k+s}^-\right)^T + \mathbf{R}_{k+s}, \tag{30b}$$

$$\mathbf{G}_{k+s}^- = \hat{\mathbf{P}}_{k+s}^- \left(\mathbf{H}_{k+s}^-\right)^T \left(\mathbf{S}_{k+s}^-\right)^{-1}, \tag{30c}$$

$$\hat{x}_{k+s} = \hat{x}_{k+s}^- + \mathbf{G}_{k+s}^-(y_{k+s} - h_{k+s}(\hat{x}_{k+s}^-)), \tag{30d}$$

$$\hat{\mathbf{P}}_{k+s} = \hat{\mathbf{P}}_{k+s}^- - \mathbf{G}_{k+s}^- \mathbf{S}_{k+s}^- \left(\mathbf{G}_{k+s}^-\right)^T, \tag{30e}$$

for $1 \leq i \leq n$ **do**
    Set $\hat{x}_{k-1}^i = \xi_{k-1}^i$ and $\hat{\mathbf{P}}_{k-1}^i = \mathbf{0}$
    Set $\hat{x}_k^i$, $\hat{\mathbf{P}}_k^i$, $\mathbf{H}_k^i$ and $\hat{h}_k^i$ using (29)–(31)
    Set $(\alpha_{k,0}^i,\ \beta_{k,0}^i,\ \mathbf{\Gamma}_{k,0}^i)$ using (32)
    for $1 \leq s \leq l$ **do**
        Set $\hat{x}_{k+s}^i$, $\hat{\mathbf{P}}_{k+s}^i$, $\mathbf{C}_{k+s-1}^i$, $\hat{c}_{k+s-1}^i$, $\mathbf{H}_{k+s}^i$, $\hat{h}_{k+s}^i$ using (29)–(31)
        Set $(\alpha_{k,s}^i,\ \beta_{k,s}^i,\ \mathbf{\Gamma}_{k,s}^i)$ using (33)–(35)
    end for
end for
Return $(\alpha_{k,l}^i, \beta_{k,l}^i, \mathbf{\Gamma}_{k,l}^i)_{i=1}^n$

Algorithm 6. Twisting function parameters for $k \geq 1$ using local EKF linearization

and then relinearizing w.r.t. $\hat{x}_{k+s}$:

$$\mathbf{H}_{k+s} = \left[\frac{\partial}{\partial x} h_{k+s}(x)\right]_{x=\hat{x}_{k+s}}, \tag{31a}$$

$$\hat{h}_{k+s} = h_{k+s}(\hat{x}_{k+s}) - \mathbf{H}_{k+s}\hat{x}_{k+s}. \tag{31b}$$

For $k = 0$, the EKF algorithm is initialized using $\hat{x}_0^- = \nu_0$ and $\hat{\mathbf{P}}_0^- = \mathbf{P}_0$ and the recursion is started from the update step (30).

The parameters $\alpha_{k,l}$, $\beta_{k,l}$ and $\mathbf{\Gamma}_{k,l}$ in (24) can be then computed recursively using the following equations. The parameters are initialized with

$$\alpha_{k,0} = \frac{\exp\{-\frac{1}{2}(y_k - \hat{h}_k)^T \mathbf{R}_k^{-1}(y_k - \hat{h}_k)\}}{|2\pi\mathbf{R}_k|^{1/2}}, \tag{32a}$$

$$\beta_{k,0} = \mathbf{H}_k^T \mathbf{R}_k^{-1}(y_k - \hat{h}_k), \tag{32b}$$

$$\mathbf{\Gamma}_{k,0} = \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k. \tag{32c}$$

Recursive updates for $1 \leq s \leq l$ are given by

$$\alpha_{k,s} = \alpha_{k,s-1} \frac{\exp\left\{-\frac{1}{2}\epsilon_{k+s}^T \mathbf{S}_{k+s}^{-1} \epsilon_{k+s}\right\}}{|\mathbf{S}_{k+s}|^{1/2}}, \tag{33a}$$

$$\beta_{k,s} = \beta_{k,s-1} + \mathbf{D}_{k+s}^T \mathbf{H}_{k+s}^T \mathbf{S}_{k+s}^{-1} \epsilon_{k+s}, \tag{33b}$$

$$\mathbf{\Gamma}_{k,s} = \mathbf{\Gamma}_{k,s-1} + \mathbf{D}_{k+s}^T \mathbf{H}_{k+s}^T \mathbf{S}_{k+s}^{-1} \mathbf{H}_{k+s} \mathbf{D}_{k+s}, \tag{33c}$$

where

$$\epsilon_{k+s} = y_{k+s} - \hat{h}_{k+s} - \mathbf{H}_{k+s} v_{k+s}, \tag{34a}$$

$$\mathbf{S}_{k+s} = \mathbf{H}_{k+s} \mathbf{K}_{k+s} \mathbf{H}_{k+s}^T + \mathbf{R}_{k+s}, \tag{34b}$$

$$\mathbf{G}_{k+s} = \mathbf{K}_{k+s} \mathbf{H}_{k+s}^T \mathbf{S}_{k+s}^{-1}, \tag{34c}$$

and the variables $\mathbf{D}_{k+s}$, $\mathbf{K}_{k+s}$ and v_{k+s} are initialized with $\mathbf{D}_{k+1} = \mathbf{C}_k$, $\mathbf{K}_{k+1} = \mathbf{Q}_k$ and $v_{k+1} = \hat{c}_k$, and then recursively computed for $2 \leq s \leq l$ using

$$\mathbf{D}_{k+s} = (\mathbf{C}_{k+s-1} - \mathbf{C}_{k+s-1}\mathbf{G}_{k+s-1}\mathbf{H}_{k+s-1})\mathbf{D}_{k+s-1}, \tag{35a}$$

$$\mathbf{K}_{k+s} = \mathbf{C}_{k+s-1}\left(\mathbf{K}_{k+s-1} - \mathbf{G}_{k+s-1}\mathbf{S}_{k+s-1}\mathbf{G}_{k+s-1}^T\right)\mathbf{C}_{k+s-1}^T + \mathbf{Q}_{k+s-1}, \tag{35b}$$

$$v_{k+s} = \mathbf{C}_{k+s-1}\left[v_{k+s-1} + \mathbf{G}_{k+s-1}\epsilon_{k+s-1}\right] + \hat{c}_{k+s-1}. \tag{35c}$$

The computational complexity of Algorithms 5 and 6 are $\mathcal{O}(nl))$. To reduce computational time, it is possible to leave

1: Set $\hat{x}_k \approx \arg\max_{x_k} p(y_{k:k+l} \,|\, x_k)$ and $\mathbf{P}_k = \mathbf{0}$
2: Set $\mathbf{H}_k$ and $\hat{h}_k$ using (31)
3: Set $(\alpha_{k,0}, \beta_{k,0}, \mathbf{\Gamma}_{k,0})$ using (32)
4: **for** $1 \leq s \leq l$ **do**
5: Set $\hat{x}_{k+s}$, $\mathbf{P}_{k+s}$, $\mathbf{C}_{k+s-1}$, $\hat{c}_{k+s-1}$, $\mathbf{H}_{k+s}$ and $\hat{h}_{k+s}$ using (29)–(31)
6: Set $(\alpha_{k,s}, \beta_{k,s}, \mathbf{\Gamma}_{k,s})$ using (33)–(35)
7: **end for**
8: Set $\alpha^i_{k,l} = \alpha_{k,l}$, $\beta^i_{k,l} = \beta_{k,l}$ and $\mathbf{\Gamma}^i_{k,l} = \mathbf{\Gamma}_{k,l}$ for all $1 \leq i \leq n$
9: Return $(\alpha^i_{k,l}, \beta^i_{k,l}, \mathbf{\Gamma}^i_{k,l})_{i=1}^n$

Algorithm 7. Twisting function parameters using EKF linearization around the mode of $p(y_{k:k+l} \,|\, x_k)$

out the relinearization of the measurement function and set $\mathbf{H}_{k+s} = \mathbf{H}^-_{k+s}$ and $\hat{h}_{k+s} = h_{k+s}(\hat{x}^-_{k+s}) + \mathbf{H}^-_{k+s}\hat{x}^-_{k+s}$. We then have $\mathbf{S}_{k+s} = \mathbf{S}^-_{k+s}$ and $\mathbf{G}_{k+s} = \mathbf{G}^-_{k+s}$, and we therefore do not need to evaluate (34b), (34c) and (35b) when computing the parameters $\alpha_{k,l}$, $\beta_{k,l}$ and $\mathbf{\Gamma}_{k,l}$. However, in our experiments, the increase in performance when using relinearization was found to clearly outweigh the increase in computational time.

B. Twisting function using linearization around the mode

The local linearization approximation requires running the EKF algorithm separately for each particle to obtain the corresponding twisting function parameters. This is computationally heavy and can make the local linearization approach too slow in practice. Computation time can be significantly reduced if we can make some assumptions about the form of $p(y_{k:k+l} \,|\, x_k)$.

The simplest case is when $p(y_{k:k+l} \,|\, x_k)$ can be assumed to be roughly symmetric and unimodal. A global approximation can be then obtained by computing the twisting function parameters using EKF linearization around the mode. This method has computational complexity of $\mathcal{O}(l)$ and is summarized in Algorithm 7.

In practice, an approximation to the location of the mode can be obtained by using a Gaussian smoother initialized from some distribution over x_k set for example as some function of the particles $(\xi^i_{k-1})_{i=1}^n$, to approximate the mean of $p(x_k \,|\, y_{k:k+l})$. We can then take the smoothed mean as an approximation for the mode. More accurate approximation of the mode can be obtained by targeting $\log p(y_{k:k+l} \,|\, x_k)$ directly and using an iterative optimization method.

For multimodal $p(y_{k:k+l} \,|\, x_k)$, the linearization could be done separately for all the modes and then combined into a mixture of exponential terms of the form in (24) (see [23] where a similar approach is used to approximate multimodal likelihoods in Gaussian mixture filters).

C. Complexity of twisted particle filters using linearization

First consider Algorithm 4 in the case that Algorithms 5 and 6 are used at lines 1 and 13 respectively. Algorithms 5 and 6 have computational complexity $\mathcal{O}(nl)$ and the full Algorithm 4 then scales as $\mathcal{O}(tnl)$.

Consider next using Algorithm 7 at lines 1 and 13 in Algorithm 4. Algorithm 7 has computational complexity $\mathcal{O}(l)$ and the overall complexity of Algorithm 4 then scales as $\mathcal{O}(t(n+l))$.

VI. Applications and numerical results

We provide here numerical examples to demonstrate the use of twisted particle filter and compare its performance against a particle filter in likelihood estimation and parameter inference using particle MCMC.

We consider the following particle filters:

- **BSPF:** bootstrap particle filter, i.e. $q_k = f_k$
- **EKFPF:** particle filter in which q_k is obtained by a standard EKF local-linearization of the importance distribution minimizing the conditional expectation of the importance weights – see [13] for details.
- **twisted-BSPF-local:** twisted version of BSPF using the EKF local linearization for the twisting function.
- **twisted-EKFPF-local:** twisted version of EKFPF using the EKF local linearization for the twisting function.
- **twisted-BSPF-mode:** computationally lighter alternative for the twisted-BSPF-local, where we use EKF linearization around an approximation for the mode of $p(y_{k:k+l} \,|\, x_k)$. For our numerical example, the approximation for the mode of $p(y_{k:k+l} \,|\, x_k)$ is obtained using an extended Rauch-Tung-Striebel (RTS) smoother [3], initialized from a Gaussian distribution over x_k, with mean and covariance given by the empirical mean and covariance of $\{c_{k-1}(\xi^i_{k-1})\}_{i=1}^n$.

We consider all the above with multinomial resampling, and also some of them with instead systematic resampling, the latter being indicated below by a suffix 'sys'.

The performance of the particle filters in likelihood estimation is measured by computing

$$\mathrm{Var}(\log Z_t) = \frac{1}{\tau}\sum_{j=1}^{\tau}(\log Z_t^j - \log \bar{Z}_t)^2, \tag{36}$$

where τ is the number of samples and $\bar{Z}_t$ is the sample mean of $\{Z_t^j\}_{j=1}^\tau$. Our interest in this quantity is that the variability of Z_t affects mixing when the particle filter is used within PMCMC. Generally speaking, higher variability degrades mixing. Probability computations are done with logarithms to avoid numerical problems.

The quality of the chain $\{\theta^j\}_{j=1}^\tau$ generated by the PMCMC algorithm can be assessed through the sample autocorrelation. Typically θ is a vector of parameters, say of length p, and the autocorrelation is computed for each 1-dimensional component θ_i, $1 \leq i \leq p$, as

$$\mathrm{ac}_i(l) = \frac{1}{\mathrm{ac}_i(0)}\frac{1}{\tau - 1}\sum_{j=1}^{\tau-l}(\theta_i^j - \bar{\theta}_i)(\theta_i^{j+l} - \bar{\theta}_i), \tag{37}$$

where l is the lag, τ is the number of samples in the chain, and $\bar{\theta}_i$ is the sample mean of $\{\theta_i^j\}_{j=1}^\tau$. Since correlations in the MCMC chain contribute to the variance of the parameter estimate, we would like to see the autocorrelation approach zero rapidly for a good quality MCMC chain.

The autocorrelation can be used to compute a single summary number for the quality of the MCMC chain, called the effective sample size [24], and given by

$$\tau_{\text{eff}} = \frac{\tau}{1 + 2\sum_{l=1}^{\infty} \text{ac}(l)}. \tag{38}$$

The effective sample size gives an approximation for the equivalent number of independent samples contained in the MCMC chain.

All the particle filters we tested were implemented in MATLAB (R2014a). Computations were performed using a MacBook Pro with 3 GHz Intel i7 and 8 Gb of memory.

A. Positioning using range and bearing measurements

The first example we consider is a target tracking problem with nonlinear measurements, where the goal is to estimate the trajectory of a moving object e.g. a vehicle or a person using range and bearing measurements from a single measurement station. This is a prototypical problem in the literature on particle filters for target tracking, see e.g., [3], [4].

The state $X = (R, V)$ consists of position $R = (R_1, R_2) \in \mathbb{R}^2$ and velocity $V = (V_1, V_2) \in \mathbb{R}^2$. The dynamical model, formed by discretizing the constant velocity continuous-time stochastic model, is linear and given by

$$X_{k+1} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t \\ \mathbf{0} & \mathbf{I} \end{bmatrix} X_k + \omega_k, \tag{39}$$

where ω_k is zero-mean Gaussian white noise with covariance

$$\mathbf{Q} = q^2 \begin{bmatrix} \Delta t^3/3\mathbf{I} & \Delta t^2/2\mathbf{I} \\ \Delta t^2/2\mathbf{I} & \Delta t\mathbf{I} \end{bmatrix} \tag{40}$$

and Δt is the time step between states. The initial state is taken to be Gaussian with mean $\nu_0 = [100, 100, 0, 0]^T$ and covariance chosen to reflect a relatively large uncertainty in the initial position,

$$\mathbf{P}_0 = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 10^2 & 0 & 0 \\ 0 & 0 & 10^{-3} & 0 \\ 0 & 0 & 0 & 10^{-3} \end{bmatrix}.$$

The measurements are the range and bearing measured from a stationary measurement station located at coordinates $(0, 0)$. The measurements are modeled by

$$Y_k = h(X_k) + \zeta_k \tag{41}$$

where

$$h(r, v) = \begin{bmatrix} \|r\| \\ \arctan(r_2/r_1) \end{bmatrix}, \tag{42}$$

and ζ_k is zero-mean Gaussian white noise, independent of ω_k, with covariance

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}.$$

The unknown parameters are the process noise variance parameter q^2 and the measurement noise variances σ_1^2 and σ_2^2. For the unknown parameters, we use independent inverse Gamma priors $\mathcal{IG}(a, b)$ with shape a and scale b parameters set to $a = b = 0.1$ for measurement noises parameters, and to $a = 1$ and $b = 0.01$ for the process noise q^2.

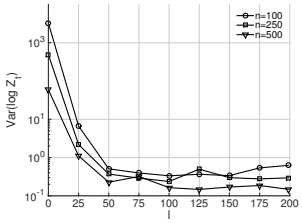


Fig. 1. Example A: Variance of $\log Z_t$ versus the parameter l for twisted-BSPF-mode-sys with different number of particles. Results are averaged over 10 datasets and 30 simulations for each dataset.

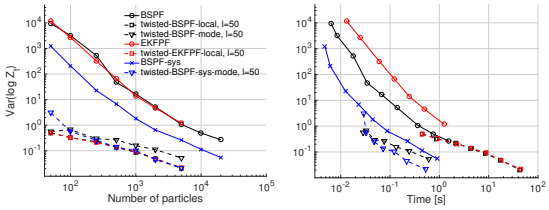


Fig. 2. Example A: Variance of $\log Z_t$ versus the number of particles n (left) and time (right) for different particle filters. Parameters are fixed to the ground truth values. Results are averaged over 10 datasets and 30 simulations for each dataset.

To test the performance of the different methods, we generated 10 datasets each consisting of $t = 200$ measurements. Fig. 1 shows the variance of $\log Z_t$ for the twisted-BSPF-mode-sys with different values of the twisting function parameter l. It can be seen that increasing the value over $l = 50$ does not give significant reduction in the variance. In the subsequent tests we fix $l = 50$.

Fig. 2 shows the variance of $\log Z_t$ for the different methods as a function of the number of particles and computation time. The twisted particle filters clearly outperform the non-twisted particle filters when looking at the $\log Z_t$ variance as a function of the number of particles. However, the local linearization based twisted particle filters have a high computation time in this example. Note also that since the EKF approximations for the importance distribution in EKFPF can be computed as a part of the local linearization for the twisting function, the computation times for twisted-BSPF-local and twisted-EKFPF-local are about the same. The twisted-BSPF-mode algorithm, based on linearizing around the mode of the twisting function, is computationally much lighter and gives the lowest variance for the $\log Z_t$ in a given computation time. For both the twisted and non-twisted particle filters using systematic resampling improves the results compared to the results using multinomial resampling.

We next analyze performance of the methods for generating samples using the Metropolis-Hastings PMCMC sampler. Based on the results in Fig. 2 we chose the twisted-BSPF-mode-sys and BSPF-sys as the test methods. We randomly chose one of the datasets as a test set and generated 20 000 samples using the PMMH sampler. An initial test run using the BSPF-sys with $n = 5000$ particles was used to tune the proposal covariance, which was then held constant for the

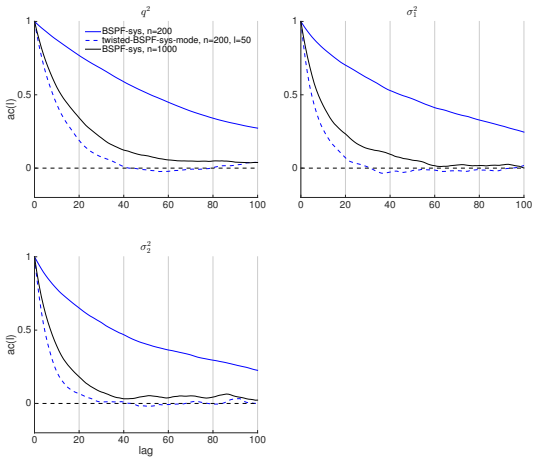


Fig. 3. Example A: Autocorrelation plots from a MCMC chain with 20 000 samples generated using PMMH with BSPF and twisted-BSPF-mode. The autocorrelations are computed with burn in of 2500 samples excluded from the computations.

TABLE I
EXAMPLE A: THE AVERAGE EFFECTIVE SAMPLE SIZES AND RELATIVE COMPUTATION TIMES FOR THE DIFFERENT PARTICLE FILTERS.

Particle filter	n	avg. τ_{eff}	rel. time
BSPF-sys	200	167.8	0.5
	500	404.6	1.0
	1000	540.6	1.7
	2000	753.5	3.4
twisted-BSPF-mode-sys	50	793.8	1.0
	100	890.0	1.2
	250	969.7	1.5
	500	1019.5	2.7

subsequent test runs.

Fig. 3 shows the autocorrelation performance for the methods. The BSPF-sys with $n = 1000$ particles has about the same computation time as twisted-BSPF-mode-sys with $n = 250$ particles and $l = 50$. The twisted-BSPF-mode-sys has clearly better autocorrelation performance than the BSPF-sys with similar computation time. The effective sample sizes and relative computation times are shown in Table I. The relative computation time is obtained as the ratio of running time for each algorithm setting to that of twisted-BSPF-sys-mode with 50 particles. Generating 20 000 samples with the twisted-BSPF-mode-sys with 50 particles took approximately 9 minutes. The twisted-BSPF-mode-sys gives clearly larger effective sample sizes in less computational time than the BSPF-sys.

The convergence of the MCMC sequence is demonstrated in Fig. 4 using normalized histograms computed from the MCMC chains. The better mixing of the MCMC chain computed using twisted-BSPF-mode-sys is especially evident in the top row histograms computed using only a small number of samples.

A simple demonstration of the tracking performance using

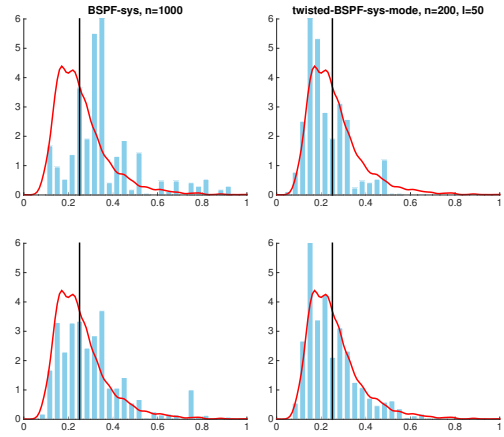


Fig. 4. Example A: Normalized histograms for parameter q^2 computed from PMCMC chains using BSPF-sys with $n = 1000$ (left) and twisted-BSPF-mode-sys with $n = 200$ and $l = 50$ (right). Number of PMCMC samples used are 500 (top row) and 2000 (bottom row). The estimated true posterior (red line) is fitted to a separate PMCMC chain with 20 000 samples. The ground truth value of the parameter is shown with a black line.

TABLE II
EXAMPLE A: TRACKING PERFORMANCE USING EKF

PMCMC method	RMSE	95% cons.
BSPF-sys, $n = 1000$	13.2	0.92
twisted-BSPF-sys-mode, $n = 200$, $l = 50$	12.4	0.91

the estimated parameter values is shown in Table II. We used an EKF algorithm with parameters fixed to mean values of the MCMC chains with 500 samples. The consistency value gives the fraction of times the true position is inside the 95% confidence ellipsoid, averaged over all the time steps. The 95% confidence ellipsoid at time k is given by

$$(\mu_k - x_{\text{true}})^T \mathbf{\Sigma}_k^{-1} (\mu_k - x_{\text{true}}) = F_{\chi_2^2}^{-1}(0.95),$$

where $F_{\chi_2^2}^{-1}(0.95)$ is the value of the χ^2 inverse cumulative distribution function with 2 degrees of freedom evaluated at 0.95.

B. Positioning using RSS measurements

As a second example, we consider estimating the parameters of a received signal strength (RSS) measurement model in an indoor positioning scenario using Bluetooth measurements. As the user moves inside the building, the positioning device measures the attenuated signal from Bluetooth base stations situated at known locations. Given a suitable model for the signal attenuation, the measurements give information about the distance between the positioning device and the base stations. Combined with a motion model, we can then use the measurements to track the user's movements inside the building.

For this example, we use a simple two parameter empirical model for the signal attenuation [25]. The base station specific

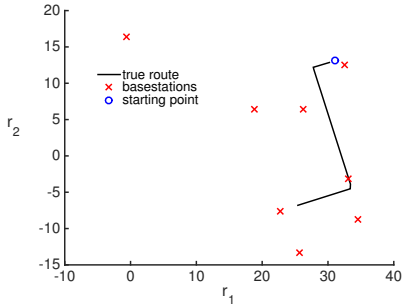


Fig. 5. Example B: The indoor positioning scenario.

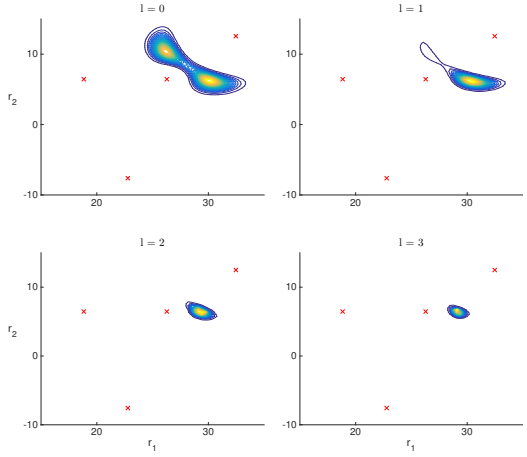


Fig. 6. Example B: Illustration of the behaviour of $p(y_{k:k+l} \,|\, x_k)$ as the number of measurements l increases. The values for $l > 0$ are computed by running a particle filter separately for each point on a dense grid for the position r_k. The velocity v_k is kept fixed for this example plot.

parameters, together with any other unknown parameters (e.g. noise variances), are estimated using a learning dataset. We consider a full Bayesian approach and use the PMCMC algorithm to draw samples from the true parameter posterior distributions. The samples can then be used to compute point estimates or integrate out the parameters in subsequent positioning phases.

We use a real data set collected in a building at the Tampere University of Technology. This data consists of RSS measurements from 8 different base stations with a total of $t = 54$ RSS measurement vectors. The locations of the base stations and the true route is shown in Fig. 5; the true route was obtained by having the user manually indicate his location on a map at regular intervals. The number of elements in the RSS measurement vector at a single time point ranges from 1 to 7, with an average number of about 5 elements per time point.

The dynamical model is the same as in the first example in Section VI-A. The initial state is taken to be Gaussian with mean ν_0 and covariance $\mathbf{P}_0$. For this example, we fix the position components of the initial mean to the true location, and set the velocity components to zero. The initial covariance is the same as in the first example.

The measurements are modelled as

$$Y_k = h_k(X_k) + \zeta_k,$$

where $h_k(x)$ is a vector with elements $h^i(x)$, $i \in \mathcal{I}_k$, where $\mathcal{I}_k$ contains the indices of the base stations whose RSS are measured at time k, ζ_k is a zero-mean Gaussian vector, independent of ω_k, with covariance $\mathbf{R} = \sigma^2\mathbf{I}$, and the RSS measurement function is [25]

$$h^i(r, v) = \rho_i - 10\lambda_i \log_{10} \|r_{\mathrm{BS},i} - r\|, \quad 1 \le i \le n_{\mathrm{BS}},$$

where $r_{\mathrm{BS},i}$ are the locations of measurement stations, λ_i and ρ_i are the base station specific parameters, and n_{BS} is the number of base stations.

The measurement likelihood is strongly non-Gaussian and can be multimodal, depending on the geometry of the base station locations. However, the term $p(y_{k:k+l} \,|\, x_k)$ becomes concentrated on a single mode as the number of measurements l increases (see Fig. 6). This allows us to reduce the computation time of the twisted particle filter by using the linearization around the mode of $p(y_{k:k+l} \,|\, x_k)$, described in Section V-B, when l is sufficiently large.

The unknown parameters are the transition noise variance parameter q^2, the process noise variance σ^2, and the measurement model parameters λ_i and ρ_i, $i = 1, \ldots, n_{\mathrm{BS}}$. Priors for the parameters are chosen as follows. For the noise variance parameters, we use independent inverse Gamma priors $\mathcal{IG}(a, b)$ with shape a and scale b parameters set to $a = b = 0.1$ for measurement noise σ^2, and to $a = 1$ and $b = 0.01$ for the process noise q^2. For the path-loss exponents λ_i, we use independent gamma priors $\mathcal{G}(a, b)$, with shape parameter $a = 3.8$ and scale parameter $b = 1.6$. For the parameters ρ_i we use independent Gaussian priors with zero mean and variance 70^2.

We first determine an initial approximation for the posterior mean by generating 10 000 samples using the PMMH and BSPF with $n = 5000$ particles. For this relatively high dimensional problem, we found that it was necessary to use a component-wise update, also called Metropolis-within-Gibbs, in the PMMH sampler. The parameters are updated in $n_{\mathrm{BS}} + 1$ blocks of 2 variables, with the blocks consisting of (λ_i, ρ_i), for $i = 1, \ldots, n_{\mathrm{BS}}$ and (q^2, σ^2) for the final block. For each block, we have an independent Gaussian random walk proposal, with covariance tuned during the initial PMMH run and kept fixed in the subsequent test runs.

Fig. 7 shows the variance of $\log Z_t$ for the twisted-BSPF-local as a function of the parameter l, with the unknown parameters fixed to the mean values from the initial test run. It can be seen that increasing l over 20 does not generally improve the results and can lead to larger variance of the estimate. This is most likely caused by the gradually increasing linearization errors in the computation of the twisting function using the EKF, meaning that for large l we have a slightly poorer approximation of the optimal twisting function. For the following tests, we use a fixed $l = 10$ for all the tested

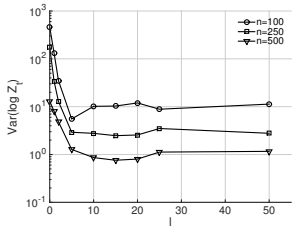


Fig. 7. Example B: Variance of $\log Z_t$ versus the parameter l for twisted-BSPF-local using multinomial resampling with different number of particles. Results are computed from 100 simulations.

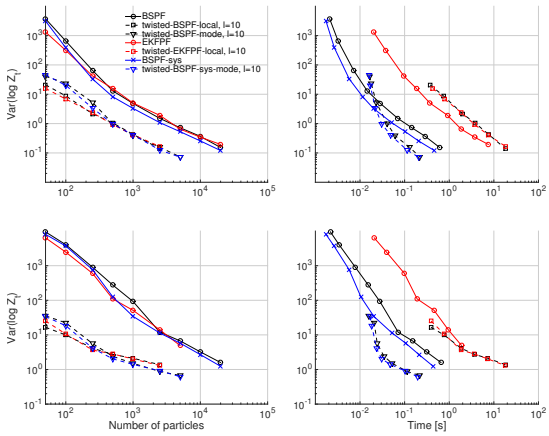


Fig. 8. Example B: Variance of $\log Z_t$ versus the number of particles n (left) and time (right) for different particle filters. Parameters are fixed to a posterior mean estimate (upper row) and to a random value chosen from test PMCMC chain (lower row). Results are computed from 500 simulations.

twisted particle filters.

Fig. 8 shows the variance of the different particle filters as a function of number of particles and computation time. Parameters were first fixed to the posterior mean estimate from the initial test run and then to a value chosen from the initial PMCMC chain, to test how the particle filters perform for parameter values away from the mean.

The results are similar as in the first example. All the tested twisted particle filters clearly outperform the non-twisted particle filters when looking at the number of particles needed for a specific log-likelihood variance. In a given computation time, the twisted-BSPF-mode-sys gives the lowest variance for $\log Z_t$. Using systematic resampling improves the results for both twisted and non-twisted particle filters.

We proceed by comparing two of the most promising particle filters, i.e. the BSPF and twisted-BSPF-mode, in generating samples using the PMMH sampler. For each particle filter, we generated a total of 100 000 samples using 10 independent chains of 10 000 samples. Fig. 9 shows the average autocorrelation plots over the 10 chains for base station parameters λ_1 and ρ_1, and noise variances σ^2 and q^2. The BSPF-sys has clearly better performance compared to the BSPF with the same number of particles, as was

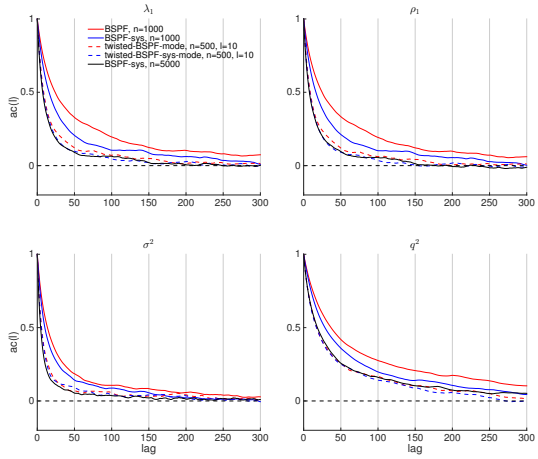


Fig. 9. Example B: Average autocorrelation plots from 10 MCMC chains generated using PMMH with BSPF and twisted-BSPF-mode. The plots are for parameters λ_1 (top left), ρ_1 (top right), σ^2 (bottom left) and q^2 (bottom right). The autocorrelations are computed from chains with 10 000 samples with burn in of 1000 samples excluded from the computations.

expected from the log-likelihood variance results. However, BSPF-sys still needs a significantly larger number of particles and longer computation times (see Table III) to reach the same autocorrelation performance as twisted-BSPF-mode.

The average effective sample sizes over all the parameters and relative computation times are shown in Table III. The relative computation time is obtained as the ratio of running time for each algorithm setting to the running time of the twisted-BSPF-mode algorithm with 250 particles. Generating 10 000 samples with twisted-BSPF-mode with 250 particles took approximately 33 minutes.

Results show that the two tested twisted particle filters give clearly the largest effective sample size with a given number of particles and in a given computational time. For the twisted particle filters, the effect of using systematic resampling is relatively small, with the systematic resampling giving slightly better results especially for a large number of particles.

A simple demonstration of the tracking performance using the estimated parameter values is shown in Table IV. We used an EKF algorithm with parameters fixed to mean values of the respective MCMC chains, computed from 10 000 MCMC samples. It should be noted that for our example, only a small amount of data was available, and for this reason the offline parameter estimation and online tracking were computed using the same data set. In reality, one would use a separate, comprehensive data set for parameter estimation.

VII. Conclusion

Our numerical results indicate that twisted particle filters can give efficiency gains for marginal likelihood approximation and parameter estimation via PMCMC. The performance gains shown in Tables I and III illustrate a speed-up of about 3-5 times for the same average effective sample size, compared to standard methods. Of course, the amount of speed-up is

TABLE III
EXAMPLE B: THE AVERAGE EFFECTIVE SAMPLE SIZES AND RELATIVE COMPUTATION TIMES FOR THE DIFFERENT PARTICLE FILTERS.

Particle filter	n	avg. τ_{eff}	rel. time
BSPF	1000	59.6	1.1
	2000	117.0	2.1
	5000	146.0	5.7
	10000	171.5	11.8
BSPF-sys	1000	81.1	0.9
	2000	124.1	1.7
	5000	165.5	4.5
	10000	191.8	9.6
twisted-BSPF-mode	250	111.0	1.0
	500	141.5	1.3
	1000	162.3	2.1
	2000	189.0	3.5
twisted-BSPF-sys-mode	250	110.0	1.0
	500	149.6	1.3
	1000	168.9	2.0
	2000	199.9	3.4

TABLE IV
EXAMPLE B: TRACKING PERFORMANCE USING EKF

PMCMC method	RMSE	95% cons.
BSPF-sys, $n = 2000$	4.6	0.09
twisted-BSPF-sys-mode, $n = 500$, $l = 10$	4.5	0.11

implementation dependent, and in our implementations we have not gone to great lengths to optimize performance of the twisted particle filter, so larger gains may well be possible. On the other hand, the efficiency of the twisted particle filter rests on the choice of the twisting functions ψ_k, and the ability to choose a "good" ψ_k is of course problem dependent.

For our purposes, a sufficient choice for ψ_k was obtained by using an EKF based linearization of the non-linear model functions. However, for problems where the EKF based methods fail to deliver a good approximation for optimal ψ_k, the presented algorithms could be modified to use linearization based on other types of Gaussian filters e.g. unscented Kalman filter or other sigma-point Gaussian filters described for example in [3].

Further research should be conducted to determine the best approach for approximating the optimal twisting function in the case of multimodal $p(y_{k:k+l} \,|\, x_k)$. A possible solution could be to use mixture approximations with each component formed by linearizing the model functions around one of the modes.

There are also various other aspects of PMCMC methodology which could be developed around twisted particle filters, for example by deriving a PMMH algorithm to sample from $p(\theta, x_{0:t}|y_{0:t})$ rather than just $p(\theta|y_{0:t})$, and in deriving particle Gibbs samplers, along the lines of those introduced in [11].

APPENDIX A
PROOF OF PROPOSITION 1

Define functions $(\eta_k)_{k=0}^t$ recursively as $\eta_t(x_t) := 1$ and $\eta_{k-1}(x_{k-1}) := \int_{\mathbb{X}} g_k(y_k \,|\, x_k) f_k(x_k \,|\, x_{k-1}) \eta_k(x_k)\, dx_k$ for $t \geq k \geq 1$. For any $1 \leq k \leq t$ we have

$$
\begin{aligned}
&\mathbb{E}\left[Z_k \frac{\sum_{i=1}^n W_k^i \eta_k(\xi_k^i)}{\sum_{i=1}^n W_k^i} \,\middle|\, \mathscr{F}_{k-1} \right] \\
&= Z_{k-1} \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n \int_{\mathbb{X}} W_k^i \eta_k(\xi_k^i) q_k(d\xi_k^i \,|\, \mathscr{L}_{k-1}^{r_{k-1}^i(U_{k-1})}) \,\middle|\, \mathscr{F}_{k-1} \right] \\
&= Z_{k-1} \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n \eta_{k-1}(\xi_{k-1}^{r_{k-1}^i(U_{k-1})}) \,\middle|\, \mathscr{F}_{k-1} \right] \\
&= Z_{k-1} \frac{\sum_{i=1}^n W_{k-1}^i \eta_{k-1}(\xi_{k-1}^i)}{\sum_{i=1}^n W_{k-1}^i}, \qquad \text{(A.1)}
\end{aligned}
$$

where the second equality follows by plugging in W_k^i and the final equality by using Assumption 1. We now have

$$
\begin{aligned}
\mathbb{E}[Z_t] &= \mathbb{E}\left[Z_0 \frac{\sum_{i=1}^n W_0^i \eta_0(\xi_0^i)}{\sum_{i=1}^n W_0^i} \right] \\
&= \frac{1}{n} \sum_{i=1}^n \mathbb{E}\left[W_0^i \eta_0(\xi_0^i) \right] = p(y_{0:t}),
\end{aligned}
$$

where the first equality follows by using (A.1) repeatedly and the final equality by plugging in W_0^i and taking the expectation.

APPENDIX B
PROOF OF THEOREM 1

It was already established in Section IV that lines 1 to 5 in Algorithm 3 draw ξ_0 from $\widetilde{\mathbf{M}}_0$. We next show that lines 11 to 17 in Algorithm 3 draw ξ_k and U_{k-1} from $\widetilde{\mathbf{M}}_k$. Plugging in $\mathbf{M}_k$ to (13b) we get

$$
\begin{aligned}
&\widetilde{\mathbf{M}}_k(d\xi_k, du_{k-1} \,|\, \mathscr{F}_{k-1}) \\
&\propto \sum_{s=1}^n \mathcal{U}(du_{k-1}) q_k(d\xi_k^s \,|\, \mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}) \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, \xi_k^s) \\
&\qquad \cdot \prod_{i \neq s} q_k(d\xi_k^i \,|\, \mathscr{L}_{k-1}^{r_{k-1}^i(u_{k-1})}). \qquad \text{(B.1)}
\end{aligned}
$$

We recognize this as a mixture form. So to sample ξ_k and U_{k-1}, we first draw the mixture component S_k on $\{1, \ldots, n\}$ with probabilities

$$
\begin{aligned}
&\widetilde{\mathcal{S}}_k(S_k = s) \propto \int_{[0,1]^m} \mathcal{U}(du_{k-1}) \\
&\cdot \int_{\mathbb{X}^n} q_k(dx_k^s \,|\, \mathcal{L}_{k-1}^{r_{k-1}^s(u_{k-1})}) \psi_k(\mathcal{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, x_k^s) \\
&\qquad \cdot \prod_{i \neq s} q_k(dx_k^i \,|\, \mathscr{L}_{k-1}^{r_{k-1}^i(u_{k-1})}) \\
&= \int_{[0,1]^m} \mathcal{U}(du_{k-1}) \\
&\qquad \cdot \int_{\mathbb{X}} q_k(dx_k \,|\, \mathcal{L}_{k-1}^{r_{k-1}^s(u_{k-1})}) \psi_k(\mathcal{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, x_k),
\end{aligned}
$$

which give the probabilities in (17) and line 11 in Algorithm 3. Next we proceed to draw U_{k-1} conditional on $S_k = s$. Given

$S_k = s$, the distribution for U_{k-1}, denoted with $\widetilde{\mathcal{U}}_{k-1}(\cdot \mid s)$, is given by

$$\begin{aligned}
&\widetilde{\mathcal{U}}_{k-1}(du_{k-1} \mid s) \\
&\propto \int_{\mathbb{X}^n} \mathcal{U}(du_{k-1}) \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, x_k) q_k(dx_k^s \mid \mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}) \\
&\qquad \cdot \prod_{i \neq s} q_k(x_k^i \mid \mathscr{L}_{k-1}^{r_{k-1}^i(u_{k-1})}) \\
&= \mathcal{U}(du_{k-1}) \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, x_k) q_k(dx_k \mid \mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}).
\end{aligned}$$

This gives (18) and line 12 in Algorithm 3. Finally, given $S_k = s$ and $U_{k-1} = u_{k-1}$, the distribution for ξ_k is proportional to

$$\begin{aligned}
&q_k(d\xi_k^s \mid \mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}) \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, \xi_k^s) \\
&\cdot \prod_{i \neq s} q_k(\xi_k^i \mid \mathscr{L}_{k-1}^{r_{k-1}^i(u_{k-1})}).
\end{aligned}$$

This gives $\widetilde{q}_k$ in (16) and lines 13 to 17 in Algorithm 3 for sampling ξ_k.

We next show that the expression for $\widetilde{Z}_k$ in Algorithm 3 can equivalently be written

$$\widetilde{Z}_k = Z_k \prod_{s=0}^{k} \phi_s, \quad k \geq 0, \tag{B.2}$$

where for each $0 \leq s \leq k$, ϕ_s is the Radon-Nikodym derivative $d\mathbf{M}_s / d\widetilde{\mathbf{M}}_s$. The result $\widetilde{\mathbb{E}}[\widetilde{Z}_k] = \mathbb{E}[Z_k]$ then immediately follows from the properties of the Radon-Nikodym derivative. Then, using Proposition 1, we get (19).

To compute the Radon-Nikodym derivatives we need to find the normalizing factors in (13a)-(13b). For $k = 0$ the normalization factor is $\int \psi_0(x) q_0(dx)$ and we get

$$\phi_0(\xi_0) = \frac{d\mathbf{M}_0(\cdot)}{d\widetilde{\mathbf{M}}_0(\cdot)}(\xi_0) = \frac{\int_{\mathbb{X}} \psi_0(x_0) q_0(dx_0)}{\frac{1}{n}\sum_{i=1}^n \psi_0(\xi_0^i)}. \tag{B.3}$$

For $k > 0$, the normalization factor is given by

$$\begin{aligned}
&\int_{[0,1]^m} \int_{\mathbb{X}^n} \frac{1}{n} \sum_{s=1}^n \mathbf{M}_k(d\xi_k, du_{k-1} \mid \mathscr{F}_{k-1}) \\
&\qquad \cdot \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, \xi_k^s) \\
&= \int_{[0,1]^m} \mathcal{U}(du_{k-1}) \\
&\qquad \cdot \frac{1}{n} \sum_{s=1}^n \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}, x_k) q_k(dx_k \mid \mathscr{L}_{k-1}^{r_{k-1}^s(u_{k-1})}) \\
&= \mathbb{E}\left[\frac{1}{n} \sum_{s=1}^n \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^{r_{k-1}^s(U_{k-1})}, x_k) \right. \\
&\qquad \left. \cdot q_k(dx_k \mid \mathscr{L}_{k-1}^{r_{k-1}^s(U_{k-1})}) \,\middle|\, \mathscr{F}_{k-1} \right] \\
&= \frac{\sum_{i=1}^n W_{k-1}^i \int_{\mathbb{X}} \psi_k(\mathscr{L}_{k-1}^i, x_k) q_k(dx_k \mid \mathscr{L}_{k-1}^i)}{\sum_{i=1}^n W_{k-1}^i} \\
&= \frac{\sum_{i=1}^n \widetilde{W}_{k-1}^i}{\sum_{i=1}^n W_{k-1}^i}
\end{aligned}$$

where we used Assumption 1 and $\widetilde{W}_{k-1}^i$ are given by (14). The Radon-Nikodym derivative for $k > 0$ is now found to be

$$\begin{aligned}
\phi_k(\mathscr{F}_{k-1}, \xi_k) &= \frac{d\mathbf{M}_k(\cdot \mid \mathscr{F}_{k-1})}{d\widetilde{\mathbf{M}}_k(\cdot \mid \mathscr{F}_{k-1})}(\xi_k) \\
&= \frac{\sum_{i=1}^n \widetilde{W}_{k-1}^i}{\sum_{i=1}^n W_{k-1}^i} \frac{1}{\frac{1}{n}\sum_{i=1}^n \psi_k(\mathscr{L}_k^i)}.
\end{aligned} \tag{B.4}$$

Writing out the expression for $\widetilde{Z}_k$ from Algorithm 3 and using the expression for Z_k from Algorithm 2, we have

$$\widetilde{Z}_0 = \frac{\sum_{i=1}^n W_0^i \int_{\mathbb{X}} \psi_0(x_0) q_0(dx_0)}{\sum_{j=1}^n \psi_0(\xi_0^j)} = Z_0 \frac{\int_{\mathbb{X}} \psi_0(x_0) q_0(dx_0)}{\frac{1}{n}\sum_{j=1}^n \psi_0(\xi_0^j)}, \tag{B.5}$$

and for $k > 0$

$$\begin{aligned}
\widetilde{Z}_k &= \widetilde{Z}_0 \frac{\sum_{i=1}^n W_k^i}{\sum_{i=1}^n W_0^i} \prod_{s=1}^k \frac{\sum_{i=1}^n \widetilde{W}_{s-1}^i}{\sum_{i=1}^n \psi_s(\mathscr{L}_s^i)} \\
&= Z_k \frac{\int_{\mathbb{X}} \psi_0(x_0) q_0(dx_0)}{\sum_{j=1}^n \psi_0(\xi_0^j)} \prod_{s=1}^k \frac{\sum_{i=1}^n \widetilde{W}_{s-1}^i}{\sum_{i=1}^n W_{s-1}^i} \frac{1}{\frac{1}{n}\sum_{i=1}^n \psi_s(\mathscr{L}_s^i)},
\end{aligned} \tag{B.6}$$

and combining (B.5)-(B.6) with (B.3)-(B.4) we observe that (B.2) holds as claimed.

Appendix C
Proof of Theorem 2

With this choice of twisting function, we have the following result for $0 \leq k \leq t-1$

$$\begin{aligned}
\widetilde{W}_k^i &= W_k^i \int_{\mathbb{X}} \psi_{k+1}(\mathscr{L}_k^i, x_{k+1}) q_{k+1}(x_{k+1} \mid \mathscr{L}_k^i)\, dx_{k+1} \\
&= W_k^i \int_{\mathbb{X}} f_{k+1}(x_{k+1} \mid \xi_k^i) p(y_{k+1:T} \mid x_{k+1})\, dx_{k+1} \\
&= W_k^i \int_{\mathbb{X}} p(y_{k+1:T}, x_{k+1} \mid \xi_k^i)\, dx_{k+1} \\
&= W_k^i\, p(y_{k+1:t} \mid \xi_k^i) = \psi_k(\mathscr{L}_k^i).
\end{aligned}$$

The final step follows by plugging in W_k^i and noting that $g_k(y_k \mid \xi_k^i) p(y_{k+1:t} \mid \xi_k^i) = p(y_{k:t} \mid \xi_k^i)$. Furthermore, for $k = t$, we have $\psi_t(\mathscr{L}_t^i) = W_t^i$.

Expanding and rearranging terms in the expression for $\widetilde{Z}_t$ in Algorithm 3 we get

$$\begin{aligned}
\widetilde{Z}_t &= \int_{\mathbb{X}} q_0(dx_0) \psi_0(x_0) \prod_{k=0}^{t-1} \frac{\sum_{i=1}^n W_k^i}{\sum_{i=0}^n \psi_k(\mathscr{L}_k^i)} \frac{\sum_{i=1}^n \widetilde{W}_k^i}{\sum_{i=1}^n W_k^i} \\
&\qquad \cdot \frac{\sum_{i=1}^n W_t}{\sum_{i=1}^n \psi_t(\mathscr{L}_t^i)} \\
&= \int_{\mathbb{X}} q_0(dx_0) \psi_0(x_0) \prod_{k=0}^{t-1} \frac{\sum_{i=1}^n \widetilde{W}_k^i}{\sum_{i=1}^n \psi_k(\mathscr{L}_k^i)} \frac{\sum_{i=1}^n W_t}{\sum_{i=1}^n \psi_t(\mathscr{L}_t^i)} \\
&= \int_{\mathbb{X}} \mu_0(dx_0) p(y_{0:t} \mid x_0) = p(y_{0:t}).
\end{aligned}$$

Acknowledgment

J. Ala-Luhtala acknowledges financial support from the Tampere University of Technology Doctoral Programme in Engineering and Natural Sciences, Emil Aaltonen foundation and KAUTE foundation. N. Whiteley and K. Heine were partly supported by EPSRC grant EP/K023330/1 and SuSTaIn.

REFERENCES

[1] N. Whiteley and A. Lee, "Twisted particle filters," *Ann. Stat.*, vol. 42, no. 1, pp. 115–141, 2014.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, 2004.

[3] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[4] B. Ristic, S. Arulampalm, and N. J. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

[5] T. B. Schön, A. Wills, and B. Ninness, "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, 2011.

[6] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models*. Springer, 2005.

[7] C. P. Casella and G. Casella, *Monte Carlo Statistical methods*. Springer, 1999.

[8] M. A. Beaumont, "Estimation of population growth or decline in genetically monitored populations." *Genetics*, vol. 164, no. 3, pp. 1139–1160, Jul. 2003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1462617/

[9] C. Andrieu and G. O. Roberts, "The pseudo-marginal approach for efficient Monte Carlo computations," *Ann. Stat.*, vol. 37, no. 2, pp. 697–725, 2009.

[10] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proc. F (Radar and Signal Processing)*, vol. 140, no. 2. IET, 1993, pp. 107–113.

[11] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *J. R. Stat. Soc. Series B (Stat. Methodol.)*, vol. 72, no. 3, pp. 269–342, 2010.

[12] C. Andrieu and M. Vihola, "Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms," *Ann. Appl. Probab.*, vol. 25, no. 2, pp. 1030–1077, Apr. 2015. [Online]. Available: http://dx.doi.org/10.1214/14-AAP1022

[13] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[14] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Am. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.

[15] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, 2002.

[16] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Process Mag.*, vol. 22, no. 4, pp. 54–69, 2005.

[17] X. Li, "RSS-based location estimation with unknown pathloss model," *IEEE Trans. Wireless Commun.*, vol. 5, no. 12, pp. 3626–3633, Dec. 2006.

[18] G. Wang, H. Chen, Y. Li, and M. Jin, "On received-signal-strength based localization with unknown transmit power and path loss exponent," *IEEE Wireless Commun. Lett.*, vol. 1, no. 5, pp. 536–539, Oct. 2012.

[19] V. Seshadri, G. V. Zaruba, and M. Huber, "A Bayesian sampling approach to in-door localization of wireless devices using received signal strength indication," in *Proc. 3rd IEEE Int. Conf. on Pervasive Comput. and Commun. (PerCom 2005)*. IEEE, 2005, pp. 75–84.

[20] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *Proc. 4th Int. Symp. Image and Signal Process. and Anal. (ISPA 2005)*. IEEE, Sep. 2005, pp. 64–69.

[21] K. B. Athreya, "Change of measures for Markov chains and the LlogL theorem for branching processes," *Bernoulli*, vol. 6, no. 2, pp. 323–338, 2000. [Online]. Available: http://projecteuclid.org/euclid.bj/1081788031

[22] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4334–4349, 2012.

[23] S. Ali-Löytty and N. Sirola, "Gaussian mixture filter in hybrid navigation," in *Proc. Europ. Nav. Conf. (GNSS 2007)*, Switzerland, May 2007, pp. 831–837. [Online]. Available: http://math.tut.fi/posgroup/aliloyttyENC2007a.pdf

[24] R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal, "Markov chain Monte Carlo in practice: A roundtable discussion," *Am. Stat.*, vol. 52, no. 2, pp. 93–100, 1998.

[25] M. Hatay, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Technol.*, vol. 29, no. 3, pp. 317–325, Aug. 1980.

Juha Ala-Luhtala received M.Sc. (Tech.) degree in information technology from Tampere University of Technology in 2011. He is currently working at IndoorAtlas Ltd. and simultaneously pursuing his Ph.D. degree in mathematics at the Tampere University of Technology. His research interests are in Bayesian inference for stochastic state-space models, with applications in positioning and navigation.

Nick Whiteley received M.Eng. and Ph.D. degrees from respectively the University of Oxford in 2004 and the University of Cambridge in 2009. He subsequently worked as a Brunel Postdoctoral fellow and Lecturer in Statistics in the School of Mathematics at the University of Bristol. His research interests are in computational statistics and especially Monte Carlo methods.

PLACE PHOTO HERE

Kari Heine received M.Sc. (Tech.) and Dr.Tech. degrees in 2003 and 2008 from the Tampere University of Technology. He has worked in private sector in software development and marketing data analysis, during 2008-2013, and as post doctoral research assistant in the University of Bristol 2013-2015. Currently he is working as a post doctoral research associate at the University College London, studying computational methods for Bayesian inference in population genetics.

Robert Piché (M'10, SM'16) received B.A.Sc., M.A.Sc. and Ph.D. degrees in 1981, 1982, and 1986 from the University of Waterloo, Canada, all in civil engineering. He was assistant professor of mathematics at cole polytechnique de Montral 1986-1988, and since 1988 has worked at the Tampere University of Technology, Finland, where he is professor since 2004. His scientific interests include mathematical modelling, numerical analysis, estimation theory, and positioning technology. He is a senior member of the IEEE.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-4091-2
ISSN 1459-2045